A Novel Data-driven Approach for Assembly System Scale-up using Simulation-based Decision Making

by

Malarvizhi Kaniappan Chinnathai

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Warwick Manufacturing Group

March 2021

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisors, Professor Robert Harrison and Dr. Daniel Vera for their invaluable advice, guidance and constant support throughout my research studies. I would also like to extend my thanks to my colleagues at Automation Systems Group, in particular Zeinab Almowafy, Mus'ab Ahmad, Hala Abdelaziz, Carlo Ferri and Rohin Titmarsh who helped and supported me in pursuing my research. Most importantly, I would like to thank my husband, Bugra Alkan for his valuable inputs and critical analysis of the thesis. I would also like to sincerely thank my parents, brother and dear grandmother who have been very caring and supportive.

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It is an original work composed by the author and has not been submitted in any previous application for any degree. The work presented (including data generated and data analysis) was carried out by the author. Over the course of PhD research, parts of this thesis have been published by the author and these publications are given in the list provided in page xi.

List of publications

Peer reviewed journal papers (Author)

M. Kaniappan Chinnathai, B. Alkan, R. Harrison. A novel data-driven approach to support decision-making suring production scale-up of assembly systems. *Accepted - Journal of Manufacturing Systems*, January 2021.

Peer reviewed journal papers (Co-author)

B. Alkan, D. Vera, M. Kaniappan Chinnathai, R. Harrison. Assessing complexity of component-based control architectures used in modular automation systems. *International Journal of Computer and Electrical Engineering*, 9(1), 393 - 402, 2017.
doi: 10.17706/IJCEE.2017.9.1.393-402
URL: http://www.ijcee.org/vol9/946-T033.pdf.

Peer reviewed conference papers (Author)

M.Kaniappan Chinnathai, B. Alkan, R. Harrison. Convertibility evaluation of automated assembly system designs for high variety production. *27th CIRP Design Conference,* 74-79, 2017. ISSN 2212-8271.
doi: https://doi.org/10.1016/j.procir.2017.01.005.
URL: https://www.sciencedirect.com/science/article/pii/S2212827117300069.

M. Kaniappan Chinnathai, T. Günther, M. Ahmad, C. Stocker, L. Richter, D. Schreiner, D. Vera, G. Reinhart, R. Harrison. An application of physical flexibility and software reconfigurability for the automation of battery module assembly. *50th CIRP Conference on Manufacturing Systems (CMS),* 604 - 609, 2017. ISSN 2212-8271.
doi: https://doi.org/10.1016/j.procir.2017.03.128.
URL: http://www.sciencedirect.com/science/article/pii/S2212827117302779.

M. Kaniappan Chinnathai, B. Alkan, D. Vera, R. Harrison. Pilot to full-scale production: A battery module assembly case study. *51st CIRP conference on Manufacturing Sytems (CMS),* 796-801, 2018. ISSN 2212-8271.
doi: https://doi.org/10.1016/j.procir.2018.03.194.
URL: http://www.sciencedirect.com/science/article/pii/S2212827118303536.

M. Kaniappan Chinnathai, Z. Al-Mowafy, B. Alkan, D. Vera, R. Harrison. A Framework for Pilot Line Scale-up using Digital Manufacturing. *52nd CIRP Conference on Manufacturing Systems (CMS),* 962 - 967, 2019. ISSN 2212-8271.
doi: https://doi.org/10.1016/j.procir.2019.03.235.
URL: http://www.sciencedirect.com/science/article/pii/S2212827119305402.

Peer reviewed conference papers (Co-author)

F.Assad, B. Alkan, M. Kaniappan Chinnathai, M.H. Ahmad, E.J. Rushforth, R. Harrison. A framework to predict energy related key performance indicators of manufacturing systems at early design phase. *52nd CIRP Conference on Manufacturing Systems (CMS),* 145 - 150, 2019. ISSN 2212-8271.
doi: https://doi.org/10.1016/j.procir.2019.03.026.
URL: https://www.sciencedirect.com/science/article/pii/S2212827119303312.

D. Kohr, M. Ahmad, B. Alkan, M. Kaniappan Chinnathai, L. Budde, D. Vera, T. Friedli, R. Harrison. Proposing a Holistic Framework for the Assessment and Management of Manufacturing Complexity through Data-centric and Human-centric Approaches. *International Conference on Complexity, Future Information Systems and Risk,* 86 - 93, 2018.
doi: 10.5220/0006692000860093.
URL: http://www.insticc.org/node/TechnicalProgram/complexis/presentationDetails/66920.

Abstract

The effect of globalisation and mass customisation necessitates that manufacturing systems respond to market trends and changes with celerity. With the increase in product variety and customisation, industries need to secure a competitive advantage over their adversaries. To realise this, it is vital to pursue a strategy to shorten the duration of the various phases of a manufacturing project. One such critical phase of the manufacturing lifecycle that has garnered relatively less attention in literature, is the scale-up phase. The main aim of this research is to propose an industrially applicable robust systematic approach to support and guide scale-up at various phases of the manufacturing system lifecycle. To fulfill this aim, it is envisioned that a two-stage Data-Driven Scale-up Model encompassing the virtual modelling and analysis of potential assembly system and workstation configurations can enable the selection of a good system design without the need to procure and commission the actual physical elements. For this purpose, the data integration of kinematic modelling tools and Discrete-Event Simulation (DES) is first explored such that the accuracy of DES input data is improved. Secondly, the approach is coupled with a multi-objective genetic algorithm optimisation module to identify assembly system designs that can lead to successful scale-up. The identified design solutions are analysed according to the pre-defined scale-up criteria and the alternate options are compared. The results of the comparison are visualised using radar charts and tables which support the decision making. An application of the DDSM framework for battery module assembly case study is provided and its benefits and shortcomings are identified.

Keywords: Scale-up, Assembly system design, Discrete-Event Simulation, Kinematic modelling, Knowledge representation, Multi-objective optimisation

Abbreviations

3D	Three Dimensional
AGV	Automated Guided Vehicle
AHP	Analytical Hierarchy Process
AMPLiFII	Automated Module to pack Pilot Line For Industrial Innovation
API	Application Programming Interface
AR	Augmented Reality
ASG	Automation Systems Group
ASRS	Automated Storage and Retrieval System
AutomationML	Automation Markup Language
BOM	Bill Of Materials
CAD	Computer Aided Design
CAEX	Computer Aided Engineering Exchange
CAM	Computer Aided Machining
CAPP	Computer Aided Process Planning
CIMOSA	Open System Architecture for Computer Integrated Manufacturing
CMSD	Core Manufacturing Simulation Data
CMSDIM	Core Manufacturing Simulation Data Information Model
COLLADA	COLLAborative Design Activity
DDSM	Data Driven Scale-up Model
DES	Discrete Event Simulation
DL	Description Language

DMS	Dedicated Manufacturing System
FDM	Factory Data Model
FIFO	First In First Out
FMS	Flexible Manufacturing System
GA	Genetic Algorithm
ICE	Internal Combustion Engine
ID	Identification
ISA	Instrumentation System and Automation
ISM	Integrated Simulation Method
IT	Information Technology
JIT	Just In Time
JT	Jupiter Tessellation
KCM	Knowledge Configuration Model
KPI	Key Performance Indicator
MATLAB	Matrix Laboratory
MCDM	Multi Criteria Decision Making
MHU	Material Handling Unit
MODAPTS	MODular Arrangement of Predetermined Time Standards
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
NPI	New Product Introduction
NSGA II	Non-dominated Sorting Genetic Algorithm - II
OPC-UA	Open Platform Communications United Architecture
OWL	Web Ontology Language
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
PPR	Product Process Resource
PPRR	Product Process Resource Resource attribute

PPRX	Product Process Resource eXchange
RFID	Radio-Frequency Identification
RMS	Reconfigurable Manufacturing System
RMT	Reconfigurable Machine Tool
SCS	System Configuration Selector
SD	System Dynamics
SISO	Simulation Interoperability Standards Organisation
STD	State Transition Diagram
SPARQL	Simple Protocol And Resource description framework Query Language
SQWRL	Semantic Query-enhanced Web Rule Language
SWRL	Semantic Web Rule Language
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
UI	User Interface
UML	Unified Modelling Language
VDSim	Virtual Driven Discrete Event Simulation
VFDM	Virtual Factory Data Model
VFF	Virtual Factory Framework
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
WCS	Workstation Configuration Selector
WIP	Work In Progress
WMG	Warwick Manufacturing Group
XML	eXtensible Markup Language

Chapter 1

Introduction

This chapter provides an insight into the motivation behind choosing the research topic, the background and formulation of the research aims and objectives. It also imparts a brief introduction to the research methodology followed by an overview of the thesis structure. The constant technological advancements in manufacturing industries, subsequently lead to the introduction of new products that kick-off projects requiring the testing and validation of both products and processes. However, the frequent changes in the customer requirements and market trends lead to uncertainty and demand fluctuations. Due to these reasons, industries need to constantly test and roll out new products and technologies at a rapid rate. Typically, this includes an initial phase of product and process validation at low volume followed by scale-up to a higher volume. To achieve successful scale-up, it is vital that industries determine strategies for smooth transition from low volume to high volume. In this regard, the aim of this chapter is to provide an understanding of the importance of scale-up phase in discrete manufacturing industries and highlight the challenges faced. A systematic approach to support decision-making during this critical phase forms the fundamental element of this research work.

1.1 Research background

The manufacturing system lifecycle, in its infant stages, comprises of the conceptual/design phase where a number of planning activities are performed. In a number of situations, the progress through the lifecycle is marked by a significant increase in product demand that necessitates the production line scale-up. The scale-up phase is a critical period during the manufacturing system lifecycle and in order to touch upon its significance, it is essential to reach a consensus about the meaning behind the term *'scale-up'*. In process and pharmaceutical industries, the term *'scale-up'* refers to an increase in the batch-size and volume of the manufacturing containers used to mix and produce the products from pilot-scale to commercial-scale [Levin, 2001; Tsinontides et al., 2004; Faure et al., 2001; Leuenberger, 2001]. In discrete manufacturing systems, *'scale-up'* refers to a significant increase in the number of finished products, either machined or assembled. Although innumerable number of guidelines and approaches have been proposed for process industry scale-up, there is a dearth of much needed literature on scale-up phase planning and implementation in the domain of discrete manufacturing industries. It is to be noted that there are significant differences between the methods and techniques used for the process or pharmaceutical industry scale-up and discrete manufacturing industry scale-up. As a result, the vast majority of knowledge that is available in the field of process industry scale-up cannot be directly translated for use in the discrete manufacturing industries. This lack of knowledge on performing efficient scale-up is evident from failed projects that result in wasted efforts, time and money.

1.2 Problem definition

In the current industrial settings, on completion of product and process validation, the next significant hurdle is to make the production line operational. If the associated technologies are fairly novel, the difficulty of the hurdle is set even higher. Due to the lack of prior knowledge about the technology, personnel experienced in that particular technology are not readily available and hence there is no clear strategy or approach to transition from low volume to high volume production. For example, consider the recent increase in the popularity of electric vehicle powertrains which is a relatively recent technology when compared to the standard Internal Combustion Engines (ICE). In this situation, the exact approach of how to scale-up a battery assembly line is a topic that needs to be given some thought. There is need for scale-up planning not only in the initial concept phase but also during operational phase (explained in more detail in section 2.4). In such situations, there is a tendency to use trial and error-based approaches to buy equipment, plan assembly system design and allocate operators. However, this approach might not be efficient in the long run and in the worst-case scenario, it might prove to be more expensive than necessary. Considering the above-mentioned issues, there is potential for formulating a methodology that is not entirely dependent on the experience of personnel and makes use of digital manufacturing to support the transition from low volume to high volume production.

1.3 Research scope

The practice of scale-up is prevalent in various industrial domains such as processing industries, pharmaceutical and discrete manufacturing systems. However, the focus of this thesis is limited to the scale-up of manual, semi-automatic and automatic discrete assembly systems. The scale-up transition is a multi-faceted procedure that requires a careful consideration of the different aspects of a manufacturing system. Some of these facets include the organisational behaviour, software domains, hardware domains, manufacturing execution systems and their integration with the production line, data communication procedures, etc. The author primarily focuses on the hardware, software and personnel modifications such as upgrading equipment, making software changes to improve functionality, increasing the number of operators, etc., to the existing production line that need to be undertaken due to the need to satisfy the new demand.

In this study, the manufacturing system is decomposed into a) component b) workstation c) pilot line d) production line and e) factory. They will, henceforth, be referred to as *'levels'*. The *'component'* level represents the highest level of granularity and a *'component'* is the basic unit of a system which can be further subdivided into elements [Lohse, 2006]. As an example, a robot is a component that is composed of elements such as motors, drives, etc. Although the robot can be decomposed into its constituent elements, the robot, as equipment, performs the necessary assemblies. Therefore, the component is not decomposed further within the research scope. The *'workstation'* level is one step higher than the *'component'* level in terms of abstraction and represents processing units that can assemble/manufacture the workpiece or product. The *'pilot line'* level which is the next higher step to the *'station'* level represents a prototype line that is used for process and product validation at low-volume. When considering production volume in-

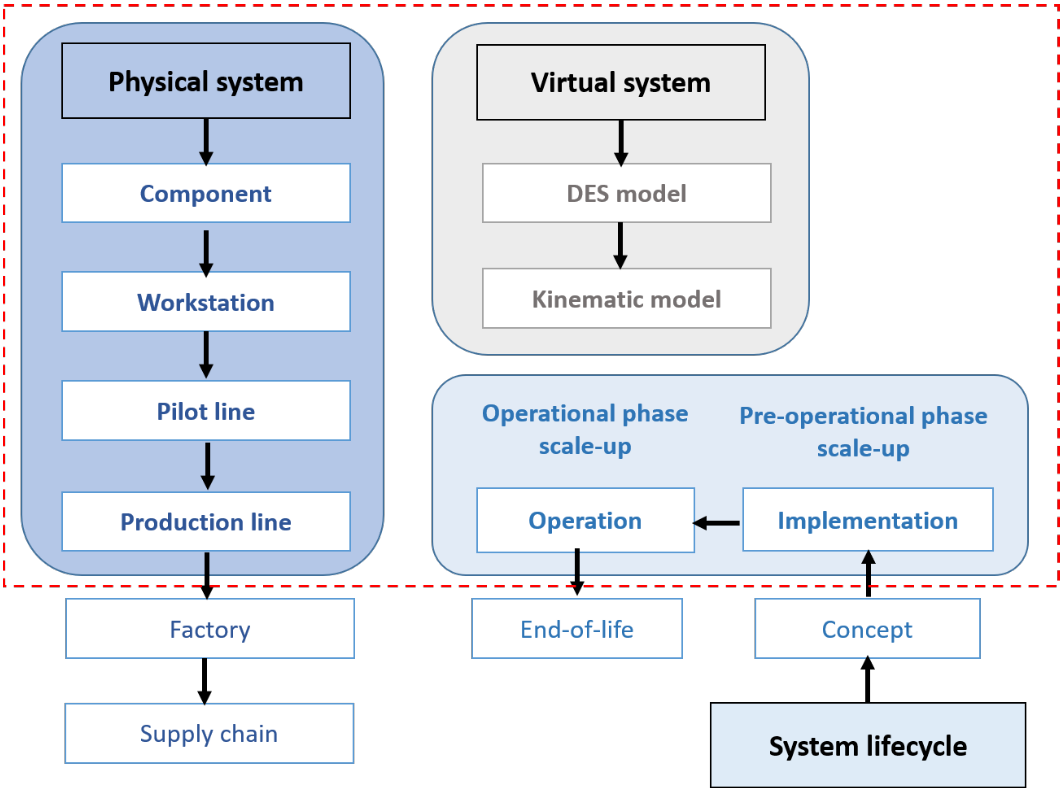


Figure 1.1: Scope of the thesis.

crease and scale-up, the term *'pilot line'* is relevant and is hence considered within the scope. It is to be noted that, in some manufacturing settings, pilot lines may not be built due to various reasons. The *'production line'* layer encompasses workstations and material handling units and is associated with assembly/manufacture of parts at a higher volume than that of the pilot line. The *'production line'* level, however, does not cover the logistics and warehouse areas. The *'factory'* level comprises of logistics, warehouses and includes one or more production lines. A manufacturing enterprise might comprise of interconnected factories as part of a *'supply chain'*. The modelling of factories, warehouses and impact of the supply chain are outside the scope of this thesis.

From **Figure 1.1**, three dimensions are considered for the research scope. In the physical system dimension, the component, station, pilot line and production line levels are considered within the scope and the factory and supply chain levels are not considered. In the virtual system dimension, the scope is limited to kinematic models and DES models. Other modelling methods such as system dynamics, agent-based modelling, multi-physics models, etc., are not investigated as part of this research. The system lifecycle dimension includes the concept, implementation, operation and end-of-life phases. As part of this research, the pre-operational phase scale-up that happens during the implementation phase of lifecycle and the operational phase scale-up that happens during the operational phase of the lifecycle are considered within the scope. More details on the flow of data among the modelling entities of the virtual system dimension and between the virtual and physical system entities are provided in **Chapter 3** of the thesis.

The intended stakeholders of the research are system designers and personnel involved in scale-up of discrete manufacturing systems that perform assembly operations. The benefits of the research can also be communicated to the higher level management using charts and graphs that allow the comparison of the considered system design solutions.

1.4 Research motivation

1.4.1 Challenges during scale-up

- **Revenue loss with increase in time-to-volume**
  The body of existing literature has acknowledged that scale-up phase is a critical one that proves to be a challenge in most situations [Tsinontides et al., 2004; Nahm and Steinfeld, 2014; Bull et al., 2008; Wirges et al., 2013]. Careful analysis of the underlying factors reveals that the degree of novelty, and product and process complexity affect the success of the scale-up projects. Industries are at a risk of losing revenue and customer satisfaction if they fail to deliver projects on time, since this phase of project overlaps with the most profitable period of a product's lifecycle. Although time-to-market is given sufficient discussion in literature, time-to-volume is not discussed in detail. In automotive, consumer electronics, and personal computer manufacturing industries, the product prices can plummet rapidly [Kurawarwala and Matsuo, 1996; Burt, June 19, 2002; Carrillo and Franza, 2006]; therefore, significant benefits can be achieved by reducing the time to volume [Terwiesch et al.,

2001].

- **Disturbances impeding the progress**

  The scale-up phase is characterised by several disturbances arising from machine stoppages, inexperienced personnel, product and process modifications, operator skill, equipment availability, material quality, equipment calibration, etc. It is crucial to limit and control these unfavourable factors as much as possible as they can lead to additional costs during the scale-up phase [Leuenberger, 2001]. Knowledge of the system at the pilot phase or low-volume production plays a significant role in optimisation of the system at a higher volume [Faure et al., 2001].

- **Impact of personnel experience on decision making**

  Scale-up is accompanied by assembly system design and technical changes [Shibasaki et al., 2006] and one main challenge during this transition is the shortage of experienced personnel; this heavily impacts the duration of the scale-up phase. From literature, it is found that relationship exists between the experience of operators and their involvement when making decisions [Karuppan and Kepes, 2006]. In novel situations, decision making is done based on inductive reasoning that involves predictions championed by the existing knowledge or experience. In such situations, inexperienced personnel can make wrong decisions or choose trial and error-based planning methods [Doltsinis et al., 2013] which could have adverse effects on the cost and time spent on the project. Additionally, the cognitive load induced by the decision making process, especially for complex manufacturing systems that involve a number of decision variables such as workstations, operators, material handling solutions, etc., can result in errors while choosing the system configurations and equipment.

The presence of such challenges hinders the success of scale-up but the existing knowledge on strategies, methods and approaches to overcome these challenges is lacking [Deif and ElMaraghy, 2007*a*]. The pitfalls associated with the implementation of scale-up that are mentioned above need to be addressed in order for industries to stay on par with their competitors. This further underscores the importance of the scale-up phase.

1.4.2 Need for framework employing simulation concepts

"Digital factory is a complete model of all the resources of a factory such as location, buildings, infrastructural media supply, logistics, machines, tools, fixtures, etc., in a standardised 3D system and managed by a factory data management system" [Westkämper, 2007]

Within the context of digital factory, it is proven that simulation and modelling helps statistical analysis of *'what-if'* scenarios and reduces the time and cost of decision making. The simulation models are also usually integrated with other Information Technology (IT) systems to support production planning and optimisation [Chryssolouris et al., 2009]. With the perceivable benefits of digital factory, the author proposes to overcome the scale-up challenges mentioned in **section 1.4.1** with the help of simulation and modelling of the manufacturing system. The use of a robust and objective framework coupled with knowledge from simulation models reduces dependency on humans in projects where no prior experience is available. Consequently, this helps reduce human errors. Moreover, integration with digital software modules is found to be beneficial to model the potential scale-up scenarios [Kuhn, 2006; Klocke et al., 2016]. This assists the development of a system design that is robust to disturbances with reduced throughput losses [Colledani et al., 2018]. Two prominent digital simulation methods, Discrete-Event Simulation (DES) and kinematic modelling are adopted to model the manufacturing system; the former will model the *'pilot'* and *'production line'* levels and the latter will model the *'component'* and *'station'* levels [Caggiano et al., 2015]. Kinematic modelling tools are typically used to support collision detection, path planning, process planning, optimal cell design and creating digital mock-up of assembly stations [Caggiano and Teti, 2018]. On the other hand, DES simulates the operational behaviour of a production line and is favourable for throughput analysis, resource utilisation and comparison of production strategies [Caggiano and Teti, 2018]. The author would like to highlight that the exchange of simulation data, presented in section 3.2.4, between the kinematic modelling software and DES software is necessary to understand and analyse complex manufacturing systems; it also supports the decision making process within DES [Caggiano et al., 2015; Ghani et al., 2015].

1.5 Research hypothesis

The framework that will be proposed is structured on the data integration of the DES models of the assembly line and kinematic models of the system workstations to improve the accuracy of input data within DES. The framework will henceforth be referred to as **'Data-Driven Scale-up Model' (DDSM)**.

'It is hypothesised that employing the DDSM framework for the transition from low-volume to high-volume production of discrete assembly systems reduces the time-to-volume and enables the selection of assembly system designs that are cost-effective and beneficial for the scale-up project.'

1.6 Research question

In discrete assembly systems, despite the criticality of the scale-up phase, there is lack of a robust framework to support it. This research study investigates the data integration and interoperability between DES and kinematic modelling software within an overarching framework that supports the scale-up phase of discrete assembly systems and answers the following question:

'How can the data integration and interoperability between kinematic and DES models for decision-making regarding the assembly system design during scale-up planning phase be achieved in a seamless way?.'

1.6.1 Research aims and objectives

- *To identify the data from the physical system/shop floor that are required by digital simulation tools, namely kinematic modelling tool and DES, which are used for modelling workstations and production lines, respectively. This is crucial to determine the type and level of data integration that would be necessary to support scale-up.*

- *To propose a robust framework for multi-domain data integration of software at two different levels of granularity, the workstation level and system level, to identify potential workstation and system configurations that can accommodate the increased capacity following scale-up. This will provide engineers* *with a decision support system that can save time, cost and effort of performing the scale-up.*

- *To demonstrate the application of the proposed methodology to support the transition from low to high volume production in a pilot line case study. This will help highlight the importance and benefits of the proposed approach.*

1.6.2 Research approach

In order to achieve the above mentioned objectives, the research approach is identified (**Figure 1.2**) and explained in the following points.

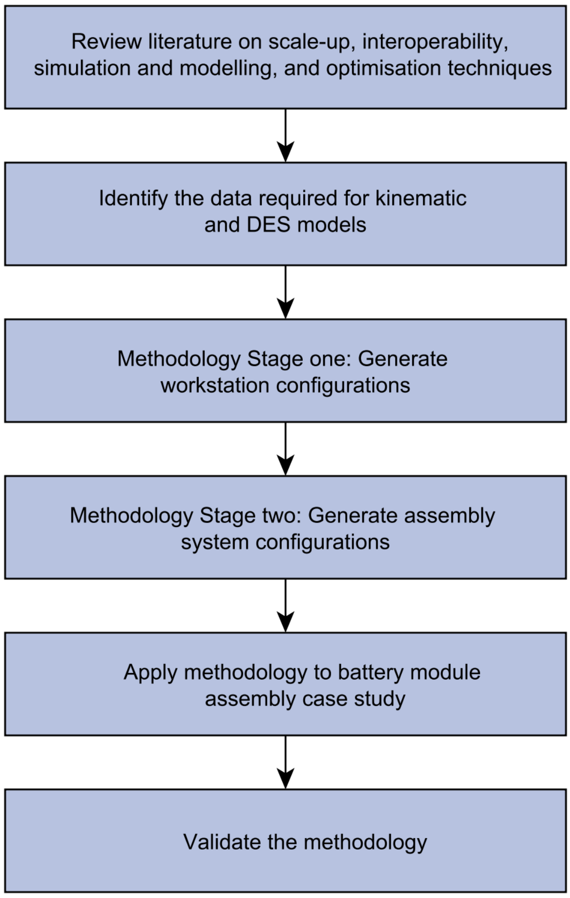


Figure 1.2: Research approach.

1. Literature review to understand the current industrial practices of scale-up and relevant knowledge existing in research articles.

2. Identification of the type, format and frequency of the data that is required by the kinematic and DES models. The required data are identified by referring to literature and focus groups with experts in the field of DES and kinematic modelling.

3. Stage one of the methodology that is associated with the workstation configuration generation is formulated. A knowledge-based kinematic model is employed for the selection of workstation equipment that can fulfill the required process.

4. Stage two of the methodology that involves the assembly line configuration generation is established. A simulation-based multi-objective optimisation approach is explored for this purpose.

5. The level and method of integration between the different modelling software such as kinematic modelling and DES software are discussed and the data structure for the integration is presented.

6. The two-stage methodology is implemented in a test case of battery module assembly and the methodology is critically reviewed.

7. The validation of the methodology to support scale-up is done by checking whether i) decision making is simplified with the help of methodology ii) the assembly system and workstation designs selected by the methodology are capable of achieving the required production volume and iii) the scale-up project time, cost and effort can be reduced. For this purpose, opinions from a focus group consisting of domain experts and system engineers are gathered.

1.7 Dissertation outline

The dissertation covers, in five chapters, the concepts of the scale-up problem, formulation of the two-stage DDSM methodology that enables assembly system design selection and decision-support using simulation based multi-objective optimisation. The various chapters in this thesis are outlined in **Figure 1.3**. The remainder of the thesis provides:

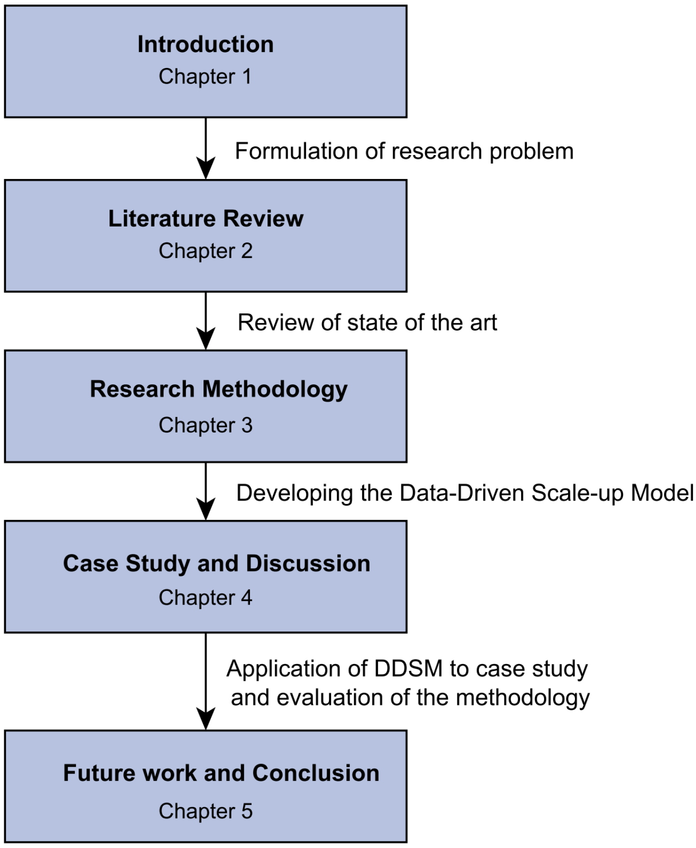


Figure 1.3: Thesis roadmap

- a detailed review of the existing approaches for scale-up, integration of data sources, and benefits and disadvantages of simulation models **(Chapter 2)**;
- a discussion of the two-stage DDSM research methodology that is built upon the two pillars of workstation configuration selection using a knowledge-based kinematic model and system configuration selection using multi-objective simulation optimisation **(Chapter 3)**;
- a demonstration of the DDSM methodology in an electric vehicle assembly setting, critical analysis and an evaluation of the presented work with certain criteria such as time, cost, effort, reusability, extendability, applicability and traceability **(Chapter 4)**;
- an assessment of whether the research objectives are fulfilled and the directions for future works **(Chapter 5)**.

Chapter 2

Literature review

The primary objective of this chapter is to provide a deeper understanding of the research topics upon which the methodology is constructed and critically analyse the relevant research works. Having provided the context of the research work, the subsequent sections provide a critical discussion of the relevant research work, highlighting the research gaps, eliciting the benefits of the DDSM methodology and comparing existing research with the DDSM framework. **Section 2.1** provides a broad description of manufacturing systems, the various economic factors that influence it and the strategies and paradigms that are associated with it. **Section 2.2**, unfolds the definitions of scalability, scale-up and capacity planning phase and sheds some light on the background of the research topic. Following this, in **section 2.3**, two specific phases of the manufacturing system lifecycle, the pre-operational and operational phases, are briefly explained to highlight the stages of the lifecycle that are associated with scale-up. Subsequently, the importance of scale-up is addressed to justify the need for research on scale-up. The difference between scale-up in a process industry and discrete manufacturing industry is also described to elicit the inability to effectively apply existing scale-up research to the problem at hand. The existing literature on ramp-up is discussed in **section 2.4**, to identify approaches and practices revolving around ramp-up that can be adapted for the scale-up phase, followed by a general discussion on the differences between the scale-up phase and ramp-up phase. An assessment of the potential of simulation and modelling to support the virtual engineering of complex manufacturing systems and its capability to aid the scale-up endeavour also forms a significant part of this chapter. A detailed review of modelling the workstations using kinematic models and assembly lines of production systems using DES is also provided in **section 2.5**. Since the modelling of complex manufacturing systems requires the data exchange among different computer-aided design and engineering software such as Solidworks, ANSYS, Catia, Process simulate, etc., a short write-up on existing frameworks for data exchange and in specific, the data integration of DES and kinematic modelling is presented in **section 2.6**. Since the latter part of the methodology adopts simulation-based optimisation for system configuration selection, **section 2.7** is dedicated to explain the basics and the rationale behind its selection. **Section 2.8** reviews the current trends in manufacturing scale-up and presents a comprehensive review of the relevant work in the area of scale-up and assembly system and workstation configuration selection. The chapter concludes by summarising the findings of the literature review which ultimately leads to the identification of research gaps.

2.1 Introduction to manufacturing systems

The term *'manufacturing system'* is used to represent the combination of manufacturing equipment, human resource, raw materials, process and information flow that enable the transformation of a product from raw material to final design. It was defined, in 1983 by CIRP (International Conference on Production Engineering) as

" a series of interrelated activities and operations involving the design, materials selection, planning, manufacturing production, quality assurance, management and marketing of the products of the manufacturing industries"

[Hitomi, 1996]

Manufacturing systems are generally divided into two types: processing and assembly. The former involves the processing of products to transform their shape, material or properties. The latter involves the joining of individual parts to form sub-assemblies or the assembly of a number of sub-assemblies to form the final assembly [Chryssolouris, 2013].

From **Figure 2.1**, the processing type manufacturing system is further classified into project shop, job shop, cellular system, continuous system and flow line. In project shop, the product or workpiece remains stationary and the required mate-

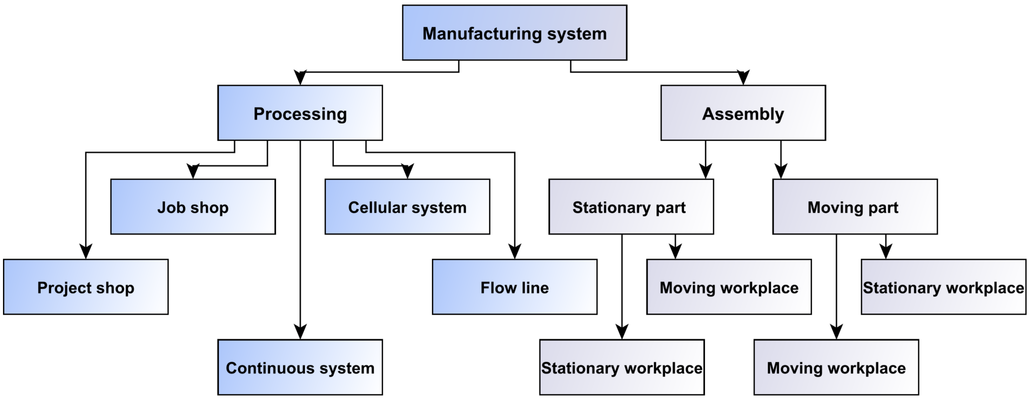


Figure 2.1: Manufacturing system classification (adopted from [Chryssolouris, 2013]).

rials, equipment and human resources are brought to it. On the other hand, in a job shop processing, similar machines and equipment are grouped together and the products move through them according to the process plan. In cellular systems, the machines that perform the required processes for a certain family of products are grouped together in a cell. Flow line processing comprises of a sequence of machines that perform the necessary processes on the product that flows through the system. The four processing types mentioned in this section are concerned with the manufacturing of discrete products. The last type of processing which is the continuous system involves products such as liquids, gases and powders. Moving onto the assembly type manufacturing, they are classified into moving part systems and stationary part systems. Both these sub-classes can be further divided depending on whether the workpiece is also moving or stationary. The stationary part systems are common in the aeronautical sector where the parts cannot be moved around easily. The moving part systems are common in the automobile sector where the raw materials and sub-assemblies are usually transported from one station to another using the material handling units [Chryssolouris, 2013].

2.1.1 Manufacturing system lifecycle

A typical manufacturing system consists of the lifecycle as illustrated in **Figure 2.2**. During the design phase, the production system is conceptualised, followed by the implementation phase where the commissioning and realisation of concepts is achieved. However, a critical hurdle that needs to be cleared to reach the operational phase is the transition to higher productivity or throughput. This transition is not a characteristic of the implementation phase alone. This demand increase can also occur in the later phases of the operational assembly system. The activity of transitioning from low-volume to high-volume is considered as scale-up and is an inseparable part of the present-day manufacturing systems. Scale-up is indeed inevitable, however, it is desirable to minimise this duration as much as possible to ensure survivability in the competitive markets.

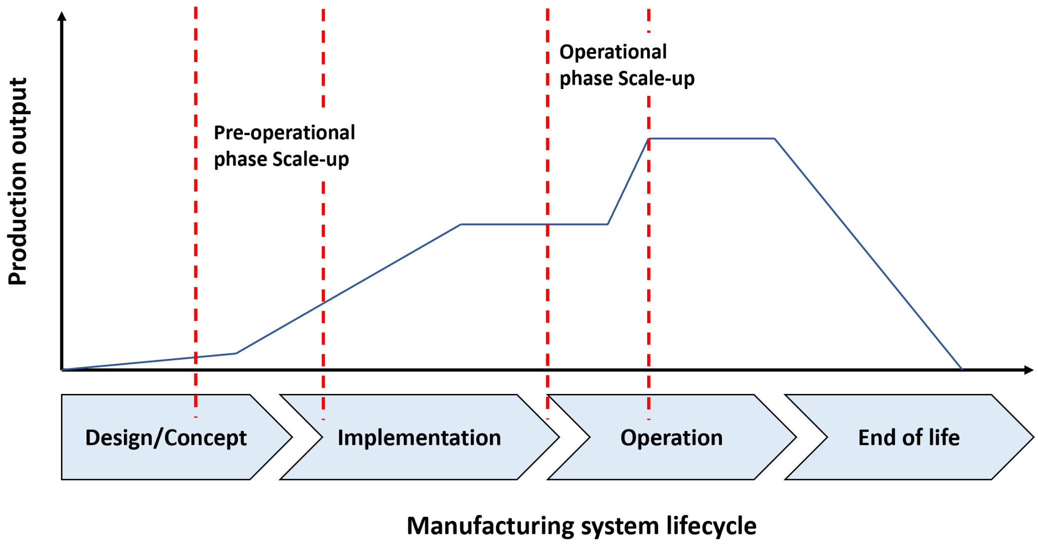


Figure 2.2: Manufacturing system lifecycle [Chase and Aquilano, 1977].

As an example, consider the case of lithium-ion battery demand surge; with the evident use of these batteries in electric vehicles, the top five battery manufacturers have been working to triple their capacity [Economist, n.d.]. Compared with the use of 17,000 electric cars in 2010, there were 7.2 million electric cars on the road in 2019. This is a booming sector which is expected to grow from 2.5 million in 2020 to 11.2 million in 2025 and reaching 31.1 million by 2030 [Woodward, 2020] and hence the need to continually increase the production capacity to meet the growing demand. In another example, the use of hydrogen powered automobiles are expected to increase and it is envisioned that by 2050, 400 million passenger vehicles, 5 million trucks, and more than 15 million buses will run on hydrogen [Up, 2017]. Compared with 2015, this could lead to a tenfold increase in the demand. In such scenarios, it is generally expected that industries have a proper procedure in place to execute such scale-up projects [Up, 2017]. However, a survey conducted by McKinsey and company in 2019 identified that industries are stuck in the pilot phase, prior to scale-up, which is termed as the *'pilot purgatory'*. The primary reasons for this were investigated and this represented in **Figure 2.3**; it was found that 45% of the respondents declared that the lack of resources and knowledge on the strategy and method to increase the production volume and the costs associated with scale-up such as planning, concept development, operator training, capital and commissioning costs, serve as major roadblocks that prevent the project's success [Garms et al., 2019]. This elicits the need for a systematic approach to support the scale-up planning and decision making.

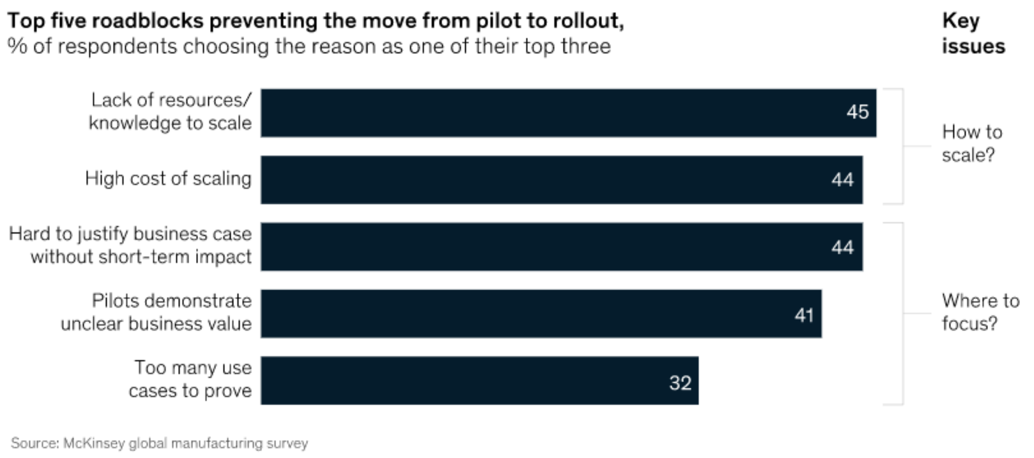


Figure 2.3: Roadblocks preventing pilot to full-scale production [Garms et al., 2019].

With the advent of Industry 4.0, manufacturing industries have started their journey towards digitalisation. Accordingly, the use of digital models to simulate the manufacturing entities is seen as an enabling technology for decision making [Mahdavi et al., 2010; Kádár et al., 2004]. The use of software that enable computer-aided manufacturing, design and engineering, allows better visualisation and embodiment of the ideas, concepts and designs pertaining to the future assembly system. This further supports the decision making process and ultimately enables faster time-to-volume. Having provided a brief introduction of the manufacturing system, its types and the necessity of scale-up, the next section describes the various manufacturing system paradigms.

2.2 Manufacturing system paradigms

Manufacturing systems have undergone several changes and paradigm shifts due to emerging technologies and market changes. Before the establishment of the existing manufacturing systems, *'craft production'* was prevalent from the 1850s. The products were tailored to the customer's needs, but they were expensive and the craftsmen were generally localised; this posed challenges for scaling up the production [Koren, 2010; Hu et al., 2011]. To overcome this issue and reduce costs associated with production, *'mass production'* paradigm comprising of assembly lines was introduced in 1900s [Hu, 2013; Ford and Crowther, 1922]. This was followed by *'lean manufacturing'* paradigm in 1955, where the idea was to reduce waste, waiting time and defects while continuously improving the system [Womack et al., 2007; Holweg, 2007]. The late 1980s saw the introduction of the *'mass customisation'* paradigm which was constructed upon the concept of Flexible Manufacturing Systems (FMS). This paradigm shift was accompanied by the huge variety and diversity of products [Hu et al., 2011; Gilmore et al., 1997]. The concept of Reconfigurable Manufacturing System (RMS) gained traction in the 1990s to enable scalable capacity and changeable functionality [ElMaraghy, 2005]. This paradigm was soon followed by *'personalised production'* in the 2000s with the advent of additive manufacturing to provide the customers with personalised unique products [Hu, 2013; Mourtzis and Doukas, 2012]. The more recent paradigm shift is towards Cloud Manufacturing, which is a service-oriented networked product development model that allows access to shared collection of distributed manufacturing resources [Wu et al., 2012, 2015]. A summary of the manufacturing system paradigms is provided in **Table 2.1**. The mass production, lean manufacturing, mass customisation, RMS, personalised production, and cloud-based manufacturing paradigms and their impact on production scale-up planning are discussed in more detail in the following sections.

Table 2.1: Evolution of manufacturing system paradigms

Year	Paradigm	Reference	Characteristic
1850	Craft production	[Koren, 2010]	High cost
			Good quality
		[Hu et al., 2011]	Localised production
1900	Mass production	[Hu, 2013]	High productivity
		[Ford and Crowther, 1922]	Low variety
			Fixed control system
			Special-purpose machines
1955	Lean manufacturing	[Ohno, 1988; Womack et al., 2007]	Reduce waste and defects
		[Holweg, 2007]	Continuous improvement
1980	Mass customisation	[Slack, 1987; Hu et al., 2011]	Catering to product variety
		[Gilmore et al., 1997]	Programmable control system
1990	RMS	[ElMaraghy, 2005]	Changeable functionality,
			Scalable capacity
			Multi-tools
2000	Personalised production	[Hu, 2013]	On-demand manufacturing
		[Mourtzis and Doukas, 2012]	Unique one-off products
2010	Cloud Manufacturing	[Wu et al., 2012, 2015]	Service-oriented network
			Distributed manufacturing

2.2.1 Mass production

The paradigm of mass production was introduced in 1900s with Henry Ford's assembly line; the main goal was to cater to the increased productivity. The systems were typically *'dedicated'* and had the ability to roll out approximately 15 million Model Ts of the same colour between 1908-27 [Williams et al., 1992]. The opinions of the customers did not matter much; they were forced to choose from the available options [Hu, 2013]. In the automotive and consumer goods industries, automated transfer lines helped the work in progress travel shorter, more direct routes. This played an important role in reducing the price of Model T from $ 950 in 1908 to $ 360 in 1916 [Williams et al., 1992; Hitomi, 2017].

The mass production systems are inherently inflexible with high capital investment and in the event of unprecedented breakdowns or machine stoppages, the whole production line is affected. This renders the system inefficient and subsequently increases the product cost [Chryssolouris, 2013]. An approach that overcomes the rigid practices of the mass production paradigm was sought after by many industries due to the intense global competition and evolving market requirements driven by the diversification of consumer preferences. For example, the number of unique vehicle designs in US increased by four-fold over a span of 35 years and the number of different running shoe styles increased from 5 to 280 over a span of twenty years [Mourtzis, Doukas, Psarommatis, Giannoulis and Michalos, 2014]. The dedicated systems used in mass production are not flexible enough to handle the highly customised products [Hu et al., 2011]. Moreover, when the demand increases, for example, by five to ten-fold, the whole transfer line needs to be replicated; this involves major changes to the hardware. However, the decision to replicate the transfer lines is not cognitively demanding on the decision maker since it just involves the replication of an entire production line. Therefore, it is possible for system designers to select assembly system designs without the guidance of a decision-support system.

2.2.2 Lean Manufacturing

The core idea behind lean manufacturing is the elimination of waste. It is also referred to as the Toyota Production System and considers several types of waste such as overproduction, excess inventory, waiting, transportation, over-processing and defects. The various concepts established as part of Lean Manufacturing include the standardisation of operation, production smoothing, Just In Time (JIT), reduction of setup time and kanban [Ohno, 1988; Monden, 2011]. It is possible to use DES to virtually represent the production system and identify wastes and non-value added activities by monitoring the resource utilisation, travel times, travel path and distance, shift times, labour utilisation and product waiting time in queue; this improves the productivity and quality [Heilala et al., 2008]. In production systems that follow lean manufacturing principles, when the system needs to be redesigned for scale-up, the decision making involves the consideration of automating the workstations, upgrading the equipment and improving the material handling in addition to replication of production lines. Since there are a number of decision variables involved in this process, it is beneficial to employ a decision-support system.

2.2.3 Mass customisation

The mass customisation paradigm became popular in the late 1980s due to the need to adapt to the changing market preferences by introducing flexibility into the system. The concepts and technologies of product family architecture and delayed differentiation are considered as key enablers for this paradigm [Slack, 1987; Hu, 2013]. However, this is also accompanied by the increase in the production system complexity with respect to the structure, operation, decision-making, etc. The FMSs play an important role in shortening the product lifecycle, time-to-market and coping with unpredictable demands [Mourtzis, Doukas, Psarommatis, Giannoulis and Michalos, 2014]. For this purpose, FMSs are beneficial, especially because they are better suited to accommodate changes than the DMSs. In FMSs, the machines, robots and equipment can be programmed to do different tasks and are not limited to producing one product type. When unprecedented events such as stoppages or machine breakdowns occur, it is still possible to continue production. However, the justification of capital expenditure is quite challenging in such systems. In general, FMSs are able to adapt to market changes and demand uncertainties but they have the problem of having more production capacity than required which could lead to a situation where the machines and system resources remain idle and are not used efficiently [Chryssolouris, 2013]. The various types of flexibilities in FMSs are: i) machine flexibility, ii) material handling flexibility, iii) operation flexibility, iv) process flexibility, v) product flexibility, vi) routing flexibility, vii) volume flexibility, viii) expansion flexibility, ix) control program flexibility, and x) production flexibility [ElMaraghy, 2005]. These different types of flexibilities provide the freedom of assembly system design while increasing the cognitive load on the decision makers. This is due to the possible combinations of solutions that can arise from the diverse range of flexibilities. Therefore, the use of a decision support system can help the system designers and decision makers in the domain of FMS scale-up.

2.2.4 Reconfigurable Manufacturing Systems

In Reconfigurable Manufacturing Systems, the machine components and material handling units can be removed, modified, added or interchanged as and when necessary. In other words, the flexibility of the system can be adjusted according to the requirements; the functionality and capacity can be varied when needed. Some of the enablers of RMSs are modular machines, standard interfaces, reconfigurable controls, sensors, adaptive controls and reconfigurable machine tools [ElMaraghy, 2005]. However, due to the addition of various modules along with their respective interfaces and integration, the RMSs exhibit increased interface, structural and operational complexity. [Wiendahl and Scholtissek, 1994]. The primary characteristics of RMSs are presented by [Koren, 2006] as modularity, integrability, customisation, scalability, convertibility and diagnosability. However, throughout the whole life, the RMSs are associated with reconfiguration costs such as incremental system capital cost, additional repeated reconfiguration and ramp-up costs [ElMaraghy, 2005]. In RMSs, when considering system design for scale-up, several design solutions are possible due to the various configurations of equipment, machine and workstations. Therefore, it is beneficial to use a decision support system for scale-up planning in such systems.

2.2.5 Personalised production

Personalised production enables tailored production of products based on the customer's preferences and needs by using technologies such as additive manufacturing, 3D printing and cyber-physical systems. The innovative products are realised by collaboration between customers and manufacturers. In this paradigm, visualisation of the design choices is very important. In parallel, it is also important to have analytical tools to evaluate the designs [Hu, 2013]. Each design can be divided into editable parts and fixed parts; the fixed parts are determined by the manufacturers to ensure required functionality. The editable parts are showcased to customers using technologies such as Virtual Reality (VR), Augmented Reality (AR) and user-friendly interfaces [Mourtzis, Doukas, Psarommatis, Giannoulis and Michalos, 2014]. The personalised productions make use of on-demand manufacturing strategies and the products are unique. In personalised production, since each product is unique, it is closely related to craft production. The personalised production paradigm is considered out of scope of the DDSM framework that will be proposed in the next chapter.

2.2.6 Cloud manufacturing

Cloud manufacturing uses network, computing and manufacturing technologies to transform manufacturing resources into services. This allows the efficient management of the resources and enables their sharing in a safe and reliable fashion across the system lifecycle. The manufacturing cloud operator imports the resources from the resource provider and exports it to the resource users. The cloud manufacturing architecture consists of five layers: i) resource layer, ii) perception layer, iii) service layer, iv) middleware layer, and v) application layer. The resource layer comprises of the manufacturing resources such as machine tools, assembly equipment, computational data, software, operators and knowledge. The perception layer is responsible for connecting the resource layer to the network. The service layer is responsible for virtualisation and encapsulation of the resources and capabilities to form service pools. The middleware layer comprises of the support services such as the energy management, failure management, cloud service management, etc. The application layer allows the users to access the cloud services [Wu et al., 2015; Zhang et al., 2014]. In cloud manufacturing paradigm, the resources are shared among different users and the operation of the manufacturing system is distributed across the cloud platforms and therefore, it is different to the other manufacturing paradigms discussed till now. This paradigm will hence not be considered for the DDSM framework.

2.3 Product Lifecycle Management

In this research, the DDSM framework will be built upon the concepts of virtualisation and digital manufacturing that subsequently allows the encapsulation of production system using virtual models. These models can be used throughout the lifecycle and can be updated as and when there are changes in the system. Since the DDSM framework supports the Product Lifecycle Management (PLM), it is necessary to discuss the existing state of PLM and their applicability in industries. The concept of PLM emerged in the early 2000s to support the lifecycle of the product and the integration of the various software tools that form part of the production process [Segonds et al., 2016]. PLM allows the management of product information and knowledge and brings innovation in product development and transfers the knowledge from industrial personnel to data management systems [Goto and Yoshie, 2019]. The family of software suites that provide functionalities intended to support the whole manufacturing lifecycle are referred to as Product Lifecycle Management (PLM) tools.

The benefits of PLM is highlighted with the case of an industrial machinery manufacturer where the various data were manually input and the Bill of Material (BOM), 2D drawings, 3D models and technical data were all stored separately. Several issues arose between the design team and manufacturing team due to the non-value added activities of redundant data input, data queries, etc. Additionally, there was dependency on operator for data retrieval; the absence of the operator delayed the whole project. With the help of PLM, however, significant improvements were noticed, especially in data organisation and queries; there was also reduced risk of losing data. Moreover, additional benefits of 3D data model distribution, reduction of manufacturing rework and associated costs were also realised [Goto and Yoshie, 2019].

Figure 2.4: Siemens PLM software overview [Sendler, 2009].

PLM has evolved over the years and the more recent cloud PLM provides digital collaboration between products and stakeholders. This removes PLM from being bound within industry premises and allows for better collaboration and reachability [Singh and Misra, 2019]. It is to be noted that aerospace firms such as Airbus and

Boeing have shown interest in PLM as a cloud service [Mas et al., 2015]. Although it is considered as an enabler for Industry 4.0, it is a relatively new concept. The industries that are already equipped with an on-premise PLM software face with the challenges of adaptability and migration to cloud PLM.

The scale-up is an activity that happens at specific phases of the product and system lifecycle. Therefore, the planning and decision making activities that need to be done are part of the product lifecycle management. There are a number of PLM tools available in the market including Enovia [Integrity, 2012], Teamcenter [Siemens, 2009], Windchill, Oracle Agile PLM and Share-A-space [Bergström and Dunford, 2007]; an illustration of the Siemens PLM software is adopted from [Siemens, 2009] and presented in **Figure 2.4**. The current PLM tools seemingly communicate in a seamless manner with each of their specialised modules. However, they are expensive and built-upon proprietary languages that do not readily integrate with other proprietary virtual modelling software that are not part of the PLM suite and enterprise information systems; they also do not meet the required industrial functionality. When establishing PLM software in industry, it disrupts the existing projects due to the installation and migration process [Sacco et al., 2010]; adopting a commercial PLM suite involves changes to the industry's processes [Vezzetti et al., 2014] and is difficult to implement and involves installation, training and maintenance costs [Hewett, 2010]. There is the need to enter a phase of transition if the industry uses a number of different software systems that have different data formats [Soto-Acosta et al., 2016]. Due to the associated costs that were discussed above and the complex adoption process, the PLM software suites may cater to the needs of large industries but fail to meet the needs of the small-scale ones [Terkaj et al., 2012].

On the other hand, the DDSM framework is not designed to be part of any specific PLM suite and hence is not restricted or bound by specific software languages. It proposes a framework that can connect with various software that may or may not be open source; the integration can be done using any database and the framework only defines the type and structure of data that needs to be integrated. The DDSM framework is intended to support the planning and decision making during scale-up phase which is only one part of the system lifecycle. The PLM suites, on the other hand, are intended to support the whole lifecycle. The DDSM framework can be employed as a low cost solution that is affordable by industries in contrast to the more expensive, proprietary solutions available in the market for the scale-up planning phase.

2.4 Scale-up

2.4.1 Defining scale-up

According to Putnik et.al, more than 1500 papers that have been published, reference the term scalability [Putnik et al., 2013], which affirms the importance of the subject matter. Koren defines the term *'scalability'* as

"the design of a manufacturing system and its machines with adjustable structure that enables system adjustment in response to market demand changes. The structure may be adjusted at the system level (e.g., adding machines) and at the machine level (changing machine hardware and control software)."

[Koren, 2006].

Scalability is regarded as a subset of reconfigurability and is closely associated with changeable manufacturing and flexibility [Putnik et al., 2013]; it is also identified as one of the characteristics of Reconfigurable Manufacturing Systems (RMS)[Koren and Ulsoy, 2002].

Closely associated with scalability is the term *'scale-up'*, originating from computer-science background, which can be defined as

"expanding a system by incrementally adding more devices to an existing node, typically by adding cpus, disks, and NICs to a node."

[Devlin et al., 1999].

The term *'scale-up'* can be interpreted in different ways depending on the context of its use. Henceforth, it is essential that it is sufficiently explored to reach consensus pertaining its definition and purpose throughout this research study. The aforementioned definition of *'scale-up'* is modified and re-defined by the author to adopt it to the manufacturing domain as *"the transition from low-volume or pilot-scale to high-volume or commercial-scale production that is realised with changes in the manufacturing system to accommodate the increase in production volume."*

The scale-up phase is characterised by modifications to the software, hardware, operator allocation and material handling units until the production volume requirements are met [Koren, 2006]. It is not possible to achieve the new demand for various products by performing minor modifications to the control strategies and manufacturing policies. Typically, the production lines need to be stopped when the line modifications are performed.

2.4.2 Difference between capacity planning and scale-up

Another term akin to the above definitions is *'capacity planning'* and it is associated with modifying the configurations of a system, both physical and logical, to accommodate demand changes [Deif and ElMaraghy, 2007*b*,*a*]. However, as seen from **Figure 2.5**, the capacity planning phase considers the daily demand change and endeavours to meet the demand, primarily, by modifying operational policies.

The capacity planning phase is characterised and influenced by the frequent but slight demand changes and modifications to the system by changes in the various operational policies such as scheduling and sequencing rules. No major hardware or software changes are generally executed and it does not comprise of production line stoppages since the scale of demand change, for example, from 300 products on day one to 320 or 290 products on day two, does not warrant such practices. On the other hand, the scale-up phase is a project that is undertaken to make major modifications to the facility, both hardware and software, that is necessitated by maybe a five-fold or ten-fold increase in demand. As highlighted from **Figure 2.5**, a sudden jump in demand from around 300 products to more than 1000 products is considered as a significant increase in demand and requires modifications to the production facility; it is represented as the scale-up phase and involves production line stoppages. **Figure 2.6** further highlights the key differences between capacity planning and scale-up.

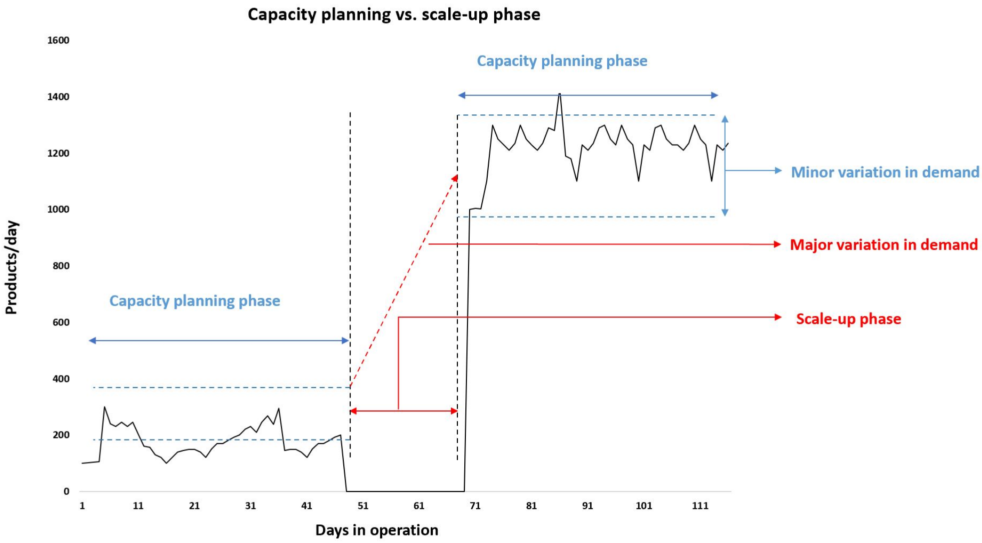


Figure 2.5: Illustration of capacity planning vs. scale-up.

2.4.3 Process industry scale-up

Scale-up in the manufacturing industry is categorised into two types: scale-up for discrete manufacturing system and scale-up for process industries. Industries that deal with pharmaceutical, food, chemical manufacturing, etc., can be regarded as process industries where the products are not discrete entities. Process industry scale-up is widely discussed in the body of literature where the term scale-up is used to refer to an increase in the manufacturing batch volume.

A notable work done by Levin explores the area of pharmaceutical scale-up [Levin, 2001]. Accordingly, the scale-up from the pilot production line involves dimensional analysis and due to the complexity of mixing and processing operations, simple extrapolation methods are not sufficient to determine the behaviour of the materials at high-volume manufacturing. Additionally, since large quantities of material are involved, there is need to give sufficient thought about the storage and material handling methods. The fundamental approach in pharmaceutical manufacturing, according to Levin, is to do a mathematical modelling of the process and perform validation at different scale-up ratios. In other related works, the scale-up of the pharmaceutical wet granulation process is discussed [Faure et al., 2001] and an approach to scale-up a pharmaceutical process is provided [Tsinontides et al.,

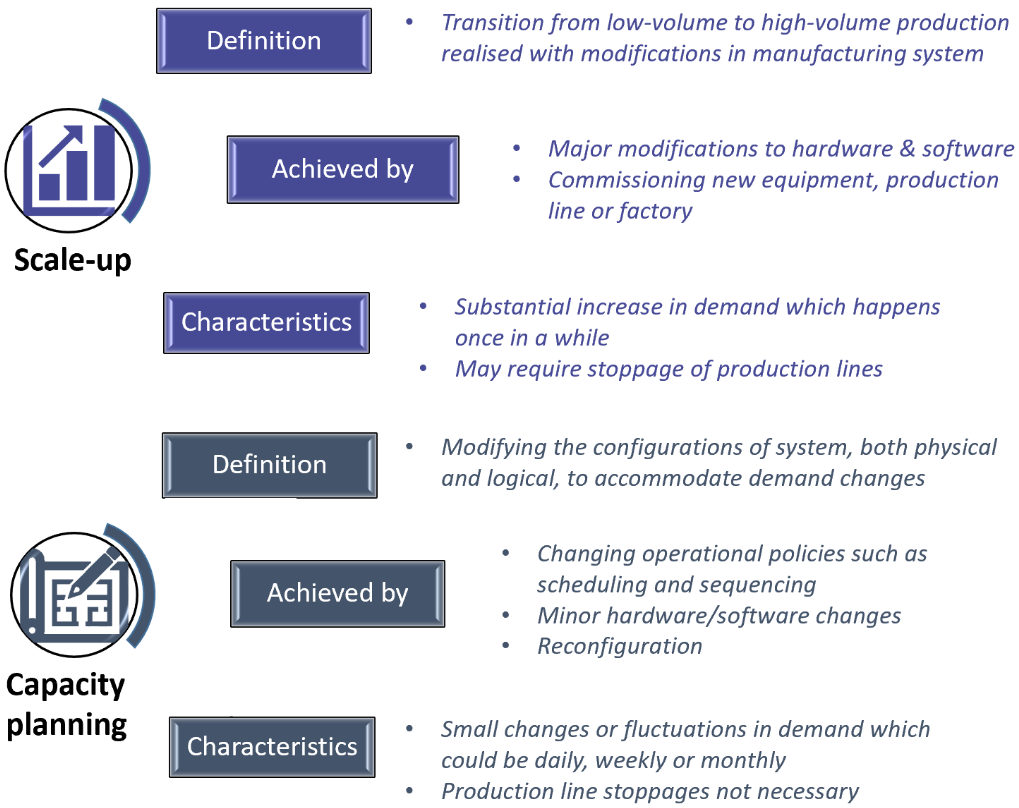


Figure 2.6: Differences between scale-up and capacity planning.

2004]. The scale-up of coating process is also analysed by Wirges et.al [Wirges et al., 2013]. From the provided references, it is understood that the fundamental differences in the operation of the process industries and discrete manufacturing makes it difficult to adopt the scale-up practices from process industries to discrete manufacturing industries. Having provided a very brief overview of the scale-up consideration in process industries, the remainder of this chapter will discuss scale-up in discrete manufacturing settings.

2.4.4 Discrete manufacturing scale-up

In discrete manufacturing, the products that are manufactured are discrete units. Therefore, there is no continuous flow of material as in process industries. Due to this reason, process industry practices such as increasing the volume of the manufacturing container for scale-up are irrelevant when it comes to discrete manufacturing. As discussed previously, in discrete manufacturing, the actual approach and method of scale-up has not been explored sufficiently. Interviews with manufacturers and white paper reviews reveal that when either new products are produced or when line is modified for volume increase, it is carried out as an ad-hoc procedure without any robust framework or systematic approach [Chemicals, 2018; Kinaxis, 2018; Kinzoku, 2008, 2018*b*, 2019; Volvo, 2007]. Conclusively, the risks associated with trial and error based methods such as choosing sub-optimal solutions, project delay or failure, delayed detection of defects, etc., are understood and their impact on the time and cost of the scale-up projects is evident [Altair, April 2021]. This further emphasizes the need for a framework to support discrete manufacturing scale-up. Henceforth, the term *'scale-up'* refers to the discrete assembly system scale-up.

2.4.5 Phases of scale-up

A typical manufacturing system lifecycle, as adapted from [Zhai et al., 2002; Jain et al., 2001], starts with the concept phase involving planning and design which leads to the implementation and commissioning of the pilot lines for prototype testing. The migration from pilot to fully operational line is an expensive process and in the case of battery assembly facilities, this is approximately 1 billion euros for an annual output of 16GWh [Volkswagen, June 2019].

Scale-up can occur at any point after the implementation of the system, but within the scope of the study, the occurrence of scale-up in two main sections, the pre-operational phase, after pilot line commissioning, and during the operational phase is considered. Scale-up includes a planning process, in which the various system configurations that satisfy the new demand are identified and compared with alternative solutions to decide the most suitable system design. It is followed by the procurement of new machines, layout changes and various modifications that need to be done to obtain the considered system design for scale-up. If new equipment or machines are procured, there might be some calibration or testing phase involved. In some situations, this might be part of a ramp-up phase. A more detailed explanation of this is provided in the following sections.

Pre-operational scale-up

Typically, early stages of manufacturing lifecycle are associated with the concept development and planning of product and process design. The validation of the conceptual design can be done either by building a pilot production line or through virtual prototyping [Wang, 2002]; this forms part of the implementation phase where concepts are implemented and tested. Virtual prototyping entails the development of the prototype model in a computer which subsequently results in cost savings of approximately 20 - 55% and time savings of approximately 40- 60% [Liu et al., 2012] [Brown and Caddick, 2003]. It has, in many aspects, partly replaced physical prototyping, however, in some situations, computational errors, image processing time delays [Wang, 2002] and issues associated with human factors and ergonomics, warrant physical prototyping. In this regard, Liu emphasises that virtual and physical prototyping complement each other [Liu et al., 2012].

The pre-operational phase physical prototyping is associated with the pilot phase of the manufacturing system lifecycle and is necessary for effective scale-up from concept/design phase to commercial scale production [Gomez and Strathy, 2001]. The pilot lines are commissioned prior to commencing the actual production run and typically, the production volume of the pilot line is very low. For example, in battery cell production this is in the range of 10 cells per day [Volkswagen, June 2019] and in battery module production this is in the range of 2 - 4 modules per day. The testing and validation of products and processes are the predominant activity; the phase primarily consists of manual operations [Butter et al., 2015].The associated benefits of pilot line include operator training, avoiding damage to equipment, determining best operating conditions, improving efficiency and ensuring safety in design and operation [*Guidebook on Design, Construction and Operation of Pilot Plants for Uranium Ore Processing*, 1990]. As the implementation phase comes to a halt, the scale-up phase is commenced. It is important to point out that the pilot production lines serve as the primary sources of data during the pre-operational phase. This pool of data, if used in a smart way to support the system, as explained in Chapter 3, could significantly reduce challenges and potential issues during the operational phase of the production line. Subsequently, the scale-up process can be better executed and the time-to-volume shortened.

Operational phase scale-up

The operational phase of a manufacturing system lifecycle commences with the production lines performing assembly operations following the pilot line testing phase and assembly line implementation or in some situations following the ramp-up phase [Almgren, 2000; Remiel et al., 2014]. During the operational phase, the production line is fully operational for commercial scale production. The customer demand during this period might stay constant or vary slightly; this might necessitate capacity planning. In some situations, due to predicted or sudden steep increase in customer demand, the need to scale-up might arise. The author refers to this phase as *'operational phase scale-up'*. In contrast with pre-operational phase scale-up, data such as processing, setup, transportation time, shift and machine breakdown, etc., are considered to be more readily available during the operational phase scale-up. Although the actual process of scale-up is similar in both situations, the system configurations identified during the operational phase scale-up planning might be more reasonable considering the availability and quality of production line data for decision making.

Table 2.2: Differences between pre-operational and operational phase scale-up.

Subject	Pre-operational scale-up	Operational scale-up
Data availability	Low	High
Connection to virtual system	Off-line	Real-time
Material handling system	Install	Upgrade
Hardware & software change flexibility	High	Low

Table 2.2 highlights the differences between the two phases of scale-up. As seen from the table, the data available during the pre-operational phase is relatively lower than the data available during operational phase. Since the pilot line is used to test the product and process and not used as an actual production line, the maintenance-related data such as first-time failure, Mean-Time Between Failures, etc., are not available. The connection to virtual models are also off-line in the case of a pilot line. On the other hand, the data availability during the operational phase is high and since the production line is completely operational, it is possible to have real-time

connection to the virtual models. When considering the material handling system, the pre-operational phase comprises of mostly manual operations with operators transporting the products between stations. Therefore, there is need to install new material handling systems. In contrast, the operational phase might already have operational material handling systems and hence they might only need to be upgraded to more productive ones. Finally, considering the flexibility of the system in terms of hardware and software changes, since in the pre-operational phase, the system is still in development, there is more freedom to make changes to the hardware and software without incurring huge cost penalties. However, in the operational phase, the hardware and software are already fixed and in operation. Therefore, identifying defects and making changes in the operational phase is difficult in terms of time and cost recovery [Altair, April 2021].

2.4.6 Importance of scale-up

Scale-up is seen as a vital element in reducing time to market and it provides industries a competitive advantage. Disturbances and issues identified during this transition can have significant impact on the efficiency of the production. The scale-up phase is characterised by high demand, low productivity and uncertainty [Haller et al., 2003]. Therefore, it is important to enable the scale-up in a smooth way to ensure that post scale-up phase can be successful [Terwiesch and Bohn, 2001]. Moreover, reduced time-to-market can enable securing more revenue before competitors since the price of high technology products, consumer goods and automotive products reduce with time [Terwiesch et al., 2001]. An example of the importance of scale-up phase for new technologies can be signposted to the recent era of powertrain electrification. The rapid rise in the demand of electric vehicles across the world has resulted in an intense competition among manufacturers to leverage the opportunity to their advantage [Kampker et al., 2017]. A study conducted in 1998 on 41 chemical-based project cases identified that it is possible to classify the projects into different categories. The projects that were most successful employed mature technologies and thorough pilot-scale testing of operations. These category one projects were able to achieve 90% capacity in six months. Category two projects took two years to reach 90% capacity and had incomplete pilot scale testing. Category three projects had an average of 80% capacity after two years but had very limited pilot testing [Mcnulty, 1998]. According to the findings of Global Lighthouse Network, 70% of manufacturers are stuck in the pilot phase without being able to progress. However, with proper procedures put in place to support the transition, productivity increases of up to 90%, 10-8% reduction in lead times, 15-20% increases in configuration accuracy and 50% increase in energy efficiency were noticeable [Betti et al., 2020]. This further underscores the need for proper scale-up planning and decision making.

2.4.7 Current industrial practices for scale-up

Despite the criticality of the scale-up phase, the current practices are not robust and systematic. This sections provides a brief summary of scale-up practices in industries. The first example is the case of a semiconductor manufacturer and as a plan for scale-up, the capacity forecasting experts determine the number of production lines required to meet new demand. Additionally, a committee formed of representatives from finance, industrial engineering, equipment engineering and construction, was given the task of reviewing the major decisions. The project execution team comprises of personnel from procurement and utilities and the project management ensures that the project is kept on course [Patel et al., 2016].

In another example, the increase in demand is achieved by changing the target cycle time. The new target cycle time is calculated and two different procedures are considered as seen from **Figure 2.7**. In procedure one, the stations that overwork are identified and identical machines are added in parallel such that the productivity is increased. However, it should be noted that the additional machines that are added in such situations also increase the need for operators and thus necessitate the procurement of automated machines or recruitment of new operators. In procedure two, the process allocations to the stations are revisited and the stations are modified and rebuilt [Tracht and Hogreve, 2012]. Following this, the new system layout is selected.
In other similar cases of scale-up, measures taken by [Kinzoku, 2018*b*] to upgrade the existing line by installing new equipment resulted in the increase of the MicroThin^{TM} capacity from 1.8 million m^2 to 2.4 million m^2. The strategy of upgrading existing line and modifying existing equipment for scale-up was pursued by

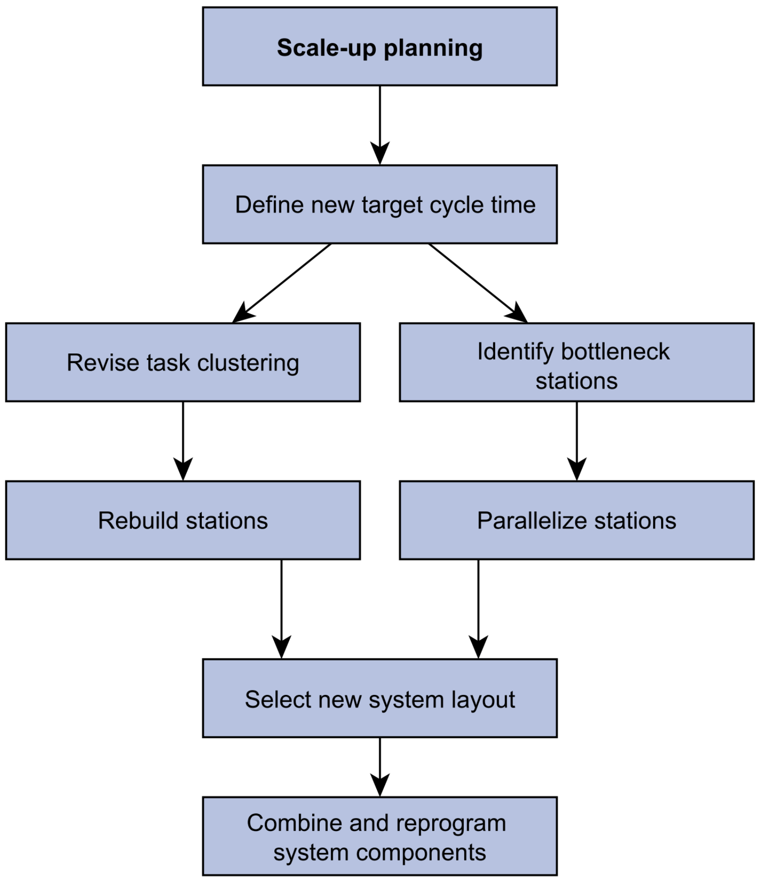


Figure 2.7: An example of scale-up practice in industry [Tracht and Hogreve, 2012].

[Kinzoku, 2019], through which the VSP® production capacity was increased from 175 tons to 420 tons per month. On a similar note, [Kinaxis, 2018] resorted to the strategy of establishing new production lines to improve their production capacity. In case of [Kinzoku, 2018*a*], a completely new production plant was established to improve the production capacity by 40%. [Chemicals, 2018] resorted to improving the level of automation and integration as their scale-up strategy. They also focused on increasing the system interoperability, which enables the communication between different software and ERP systems. [Volvo, 2007] decided to build new plants and production equipment along with increasing the degree of automation as their strategy. [Nazzal et al., 2006] added more machines to improve productivity, whereas [Venkataraman et al., 2014] focused on process improvement, modification and tooling change. A summary of these scale-up practices is provided in **Figure 2.8**.

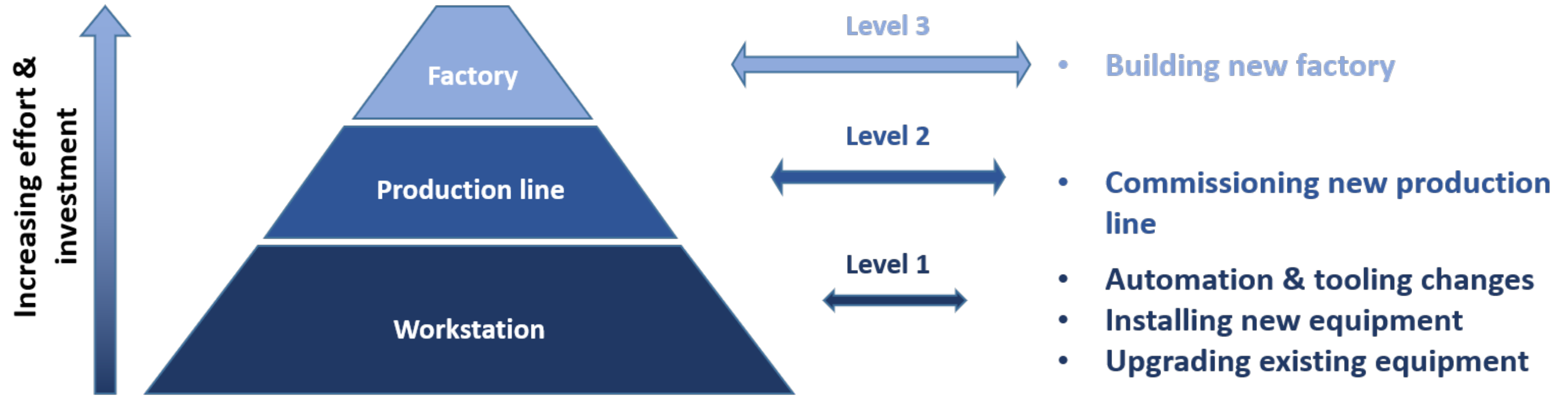


Figure 2.8: Summary of scale-up practices in industry.

The above-mentioned measures or practices for scale-up could be the result of decisions made by respective personnel based on experience or intuition. Experience-based decision making is where the personnel who have significant experience in system design and process improvement make decisions based on their previous experiences. However, if the organisation does not have an experienced engineer or manager, there is a risk that the decisions taken could actually prolong the scale-up phase due to unprecedented errors. Similarly, if the technology, process or product is new then there is no prior experience in handling such situations. Therefore, there is the risk of making wrong decisions. Apart from experience-based decision making, there is the possibility of simulation-based decision making for scale-up where the scale-up decisions are not made as an ad-hoc activity. The simulation-based decision making is discussed in more detail in Chapter 3.

2.5 Ramp-up knowledge applicable for scale-up

Ramp-up phase is defined as

"the time between the first part produced following system reconfiguration until reaching the required throughput level."

[Colledani et al., 2018].

The ramp-up phase in early-design stages is characterised by transition from development to commercial scale production; main tasks in ramp-up involve achieving the required level of quality, cost and throughput [Elstner and Krause, 2014]. In literature, the terms *'ramp-up'* and *'scale-up'* have been used interchangeably and hence this section is intended to provide more clarity and consistency in the use of the terms. In the manufacturing system lifecycle, ramp-up phase commences on conclusion of the implementation stage where process conception and development is done [Slamanig and Winkler, 2011] and is primarily associated with New Product Introduction (NPI) and product design modifications.

On the other hand, scale-up phase is not necessarily tied to NPI. Depending on the industry and the phase of the system lifecycle, scale-up phase may or may not be pursued by a ramp-up phase. While the term ramp-up considers product volume, variety and quality and commences on completion of the planning activities for

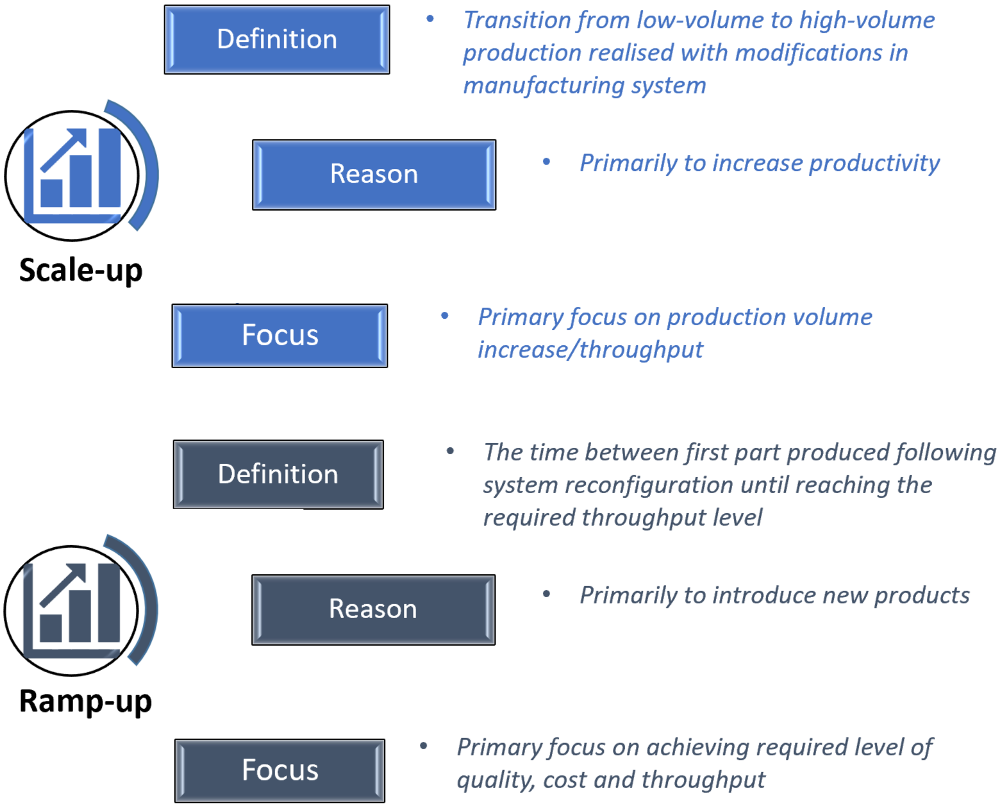


Figure 2.9: Differences between scale-up and ramp-up.

major system modifications and ends on achieving the desired targets, the term scale-up primarily considers product volume increase. The key differences between ramp-up and scale-up are represented in **Figure 2.9**.

It is to be noted that ramp-up phase might be prolonged with additional adjustments to meet the target if the planning phase involves poor decision making. Since both scale-up phase and ramp-up phase intend to achieve the desired volume, it is envisioned that the available research and knowledge on ramp-up could be applicable for scale-up. Hence the review on ramp-up as part of the literature survey.

2.5.1 Review of relevant work

This section identifies the ramp-up research work which can potentially be applied to the considered research problem. Stauder [Stauder et al., 2014] identified a framework to assess whether a set of selected technologies could provide the required process capability following ramp-up. The framework provides decision support for technology selection in the context of high volume production and is the first published research for detailed technology selection during ramp-up. This concept proved beneficial for formulation of Stage one workstation configuration selection of the DDSM framework. The second research work that is discussed is by [Klocke et al., 2016], who proposed a framework where a hybrid simulation model using DES and system dynamics simulation was used to support the ramp-up phase. In their research, they identified that the various factors affecting the dynamics of ramp-up include disturbances such as machine tool breakdowns, improvement measures, and process variability from worker capability, machine tool capability, and planning uncertainty and adjustments. With the focus on the machine tool breakdown, the impact of the manufacturing technology on the ramp-up was analysed as part of their research. This helped develop the workstation configuration selection process for the DDSM framework. In another related work, [Surbier et al., 2014] summarised available literature pertaining to ramp-up in which they mentioned the characteristics of ramp-up phase and the problems faced during ramp-up. This provided insights into the type of problems and disturbances that might extend the time-to-volume.

A simulation-based approach to plan for personnel during ramp-up was discussed as part of another research, where an algorithm using Plant Simulation was employed and DES-based decision support during ramp-up phase for planning the human resource was proposed [Lanza and Sauer, 2012]. This underscores the benefits of using DES models for decision making during the ramp-up phase and subsequently, the scale-up phase. According to [Colledani et al., 2018], the anticipation of disturbances that affect the system can lead to reduction of throughput losses during ramp-up by creating a system design that is robust. Moreover, strategies to improve the efficiency of production ramp-up is proposed as part of their work. From this research, the potential of data analytics, digital manufacturing, communication standards and on-line data acquisition to reduce the time-to-volume and improve the efficiency of the ramp-up phase is evident. In another piece of work, three performance metrics, functionality, quality and performance optimisation were discussed to measure the progress of ramp-up [Doltsinis et al., 2013]. These performance measures and their significance in determining the success of the ramp-up phase are considered important and can be adopted for the comparison of alternate system designs in the DDSM framework. This concludes the review of relevant work and the next section explains simulation and modelling which is one of the key enabling technologies for scale-up.

2.6 Enabling technology for scale-up

2.6.1 Simulation and modelling

"Simulation modelling and analysis is the process of creating and experimenting with a computerized mathematical model of a physical system"

[Chung, 2003]

The history of simulation can be traced back to the introduction of Monte-carlo simulation in 1777. From this period onwards, the evolution of simulation and its use across various domains is presented by Mourtzis in their article [Mourtzis, Doukas and Bernidaki, 2014; Mourtzis, 2020*a*]. The first general purpose simulation model was created in 1960 by Tochter and Owen under the General Simulation Program to simulate an industrial plant [Tocher and Owen, 2008]. From then on, simulation and modelling activities gained traction and several approaches and techniques for the same were proposed. It found widespread use in a diverse range of domains including healthcare, manufacturing, service, military, telecommunication and transportation [Fishman, 2013]. Specifically, in manufacturing systems, simulation and modelling can be employed at various levels of abstraction for modelling the machining, assembly, material handling and logistics, human resource modelling, thermodynamics, product flow and warehousing. Concurrently, service-based simulations are employed in healthcare, food and entertainment, information technology, and retail stores. In the field of transportation, simulation and modelling is used to model airport, train, bus and logistics [Chung, 2003].

The basic elements of a digital factory are identified as i) construction, that represents the mechanical or construction information such as dimensions or connector types, ii) function, that represents the operating functions or tasks, iii) performance, that represents the cycle time, energy consumption, etc., iv) location, that represents the absolute, relative or global location and v) business, that represents the demand, delivery time, etc. The relationship between these elements can be structural or operational, permanent or temporary [BSI, 2016]. The cornerstone of digital factory is the use of virtual engineering to simulate and model the stations, machines, robots, and the control logic of a production line and allow data transfer in a seamless way. The modelling and simulation of the manufacturing systems is considered as a very effective tool to experiment and validate product, process and systems before the production systems become operational.

Simulation is even more useful in current days to model the complex industrial scenarios [Mourtzis, Doukas and Bernidaki, 2014]. In specific, simulation and modelling is found effective in testing new strategies, comparing alternate solutions and understanding more about the system-at-hand. For this purpose, a myriad of simulation tools and methods ranging from CAD, physics-based modelling, robot path planning, process and kinematic modelling, layout optimisation, assembly line balancing, capacity planning, scheduling and resource allocation exist [Jahangirian et al., 2010]. Kuhn has highlighted the reasoning behind the need for an integrated digital factory using virtual modelling tools [Kuhn, 2006]; the key advantages are highlighted as the reduction in time-to-market and time-to-benefit.

According to [Law, 1986], a typical simulation study generally consists of the following steps:

1. The problem is formulated and objectives are defined. The criteria for comparing system designs are specified and the cost and time of study is investigated.

2. The relevant data are collected and the model is defined. Since the accuracy of the simulation results depend on the input data, a lot of importance needs to be given to this step. The input data can come from various sources such as time studies, historical records, supplier documents, intuition and experience.

3. The statistical distributions and stochasticity are modelled in this step. An example is the machine failure modelling where the breakdown and repair times are indicated as statistical distributions.

4. After the model is built with necessary data and distributions, it needs to be verified to check that the model is constructed the way it was intended to. The model also needs to be validated to check if it represents the system that is modelled.

5. The experiment needs to be designed by considering the various scenarios and the value of variables for those scenarios. Additionally, experimentation parameters such as the number of simulation runs, warm-up time, etc., need to be defined

6. The last step is to analyse the value of the performance measures obtained from the experimentation, validate and verify the simulation results, and perform objective-specific studies.

2.6.2 Merits and demerits of simulation and modelling

There are various reasons for employing simulation in production systems. Simulation models provide the benefit of experimentation in shorter periods of time and have the ability to expand or shorten the time period to understand the dynamics of the considered system. Therefore, various analyses can be performed in an efficient way. Prior to the establishment of dynamic simulations, systems were studied as static models. However, in recent days, the flexibility offered by simulation tools allows modelling of systems with more accuracy and realism. With simulation tools, the analysis of manufacturing systems has become easier as the user does not need to worry about the calculations and analytics that run in the background. For example, the use of robot path planning software enables the user to plan the path of the robot using either forward or inverse kinematics without the need to actually do the calculations regarding the orientation and displacement. Moreover, simulation allows visualising the manufacturing system dynamics and is paramount for management decision making [Chung, 2003; Fishman, 2013; Bangsow, 2012]. Moreover, in early stages of production systems, due to the absence of the physical assembly line, the various design formulations and ideas need to be bolstered by the virtual models that can help sharpen the understanding of the system. This enables the comparison of various alternatives in the virtual environment which is less expensive than building physical models.

However, on the downside, simulation heavily relies on the quality of input data; lack of high quality input data adversely impacts the simulation results. In some simulation software such as DES, specialised training and understanding of statistics is important [Chung, 2003]. In addition to the initial licensing cost of simulation tool, the training cost, maintenance upgrade costs, hardware cost, data acquisition cost, translation of the company data cost, system integration cost and database cost play an important role in limiting its implementation in manufacturing industries [McLean and Leong, 2001]. The process of modelling requires data and understanding of the production system that is being modelled. Running the simulation model becomes time-consuming and computationally demanding with the increase in the number of replications and model details [Brailsford, 2014]. The use of simulation in industrial settings is affected by the cost of simulation software; the simulation affordability is affected by the availability of experienced staff, information system infrastructure, complexity of the application area, availability and format of input data.

2.6.3 Use of simulation across manufacturing system lifecycle

The simulation and modelling techniques can be used across the lifecycle of a manufacturing system and do not have to be limited to a specific phase of the lifecycle. Digital tools are used across the lifecycle for activities ranging from product design analysis to data management and optimisation of resources [Camba et al., 2017]. Due to these benefits, the migration from a conventional system to a model-based enterprise is understandable. It has been highlighted by [Bishop, 2015] that the advantages of simulation can be exploited when the simulation models are used throughout the lifecycle for various applications such as supporting decision making, performance prediction, testing, evaluation, etc. This is also reflected in the concept of Model-Based System Engineering.

"Model-Based System Engineering is the formalized application of modeling to support system requirements, design, analysis, verification and validation, beginning in the conceptual design phase and continuing throughout development and later lifecycle phases."

[INCOSE, 2007]

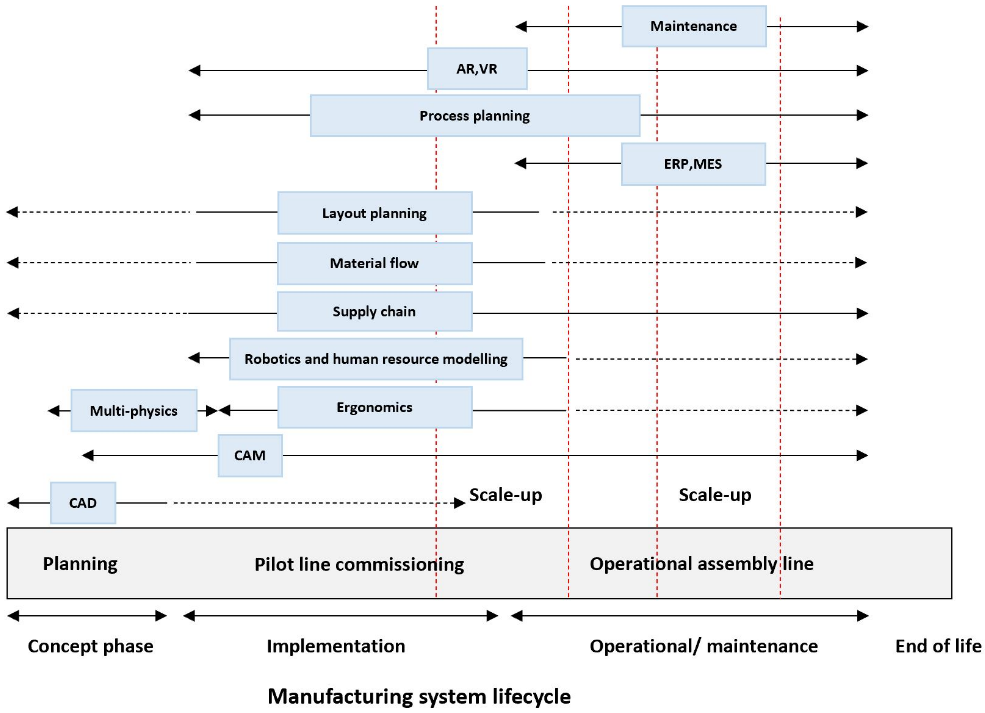


Figure 2.10: Usage of simulation and modelling tools across the lifecycle.

Figure 2.10 shows the author's work and interpretation of the usage of simulation tools across the lifecycle, starting from the concept phase. The solid lines in the Figure show the phase of lifecycle where the simulation software are predominantly used for a particular application. Although the use of simulation tools, the type of method and duration of usage across the lifecycle depends on the application and industry, **Figure 2.10** is used to provide a general understanding of the concept and is adapted from research done by Mourtzis [Mourtzis, Doukas and Bernidaki, 2014]. In their work, Mourtzis elicit that simulation tools can be used for various applications such as i) CAD, ii) CAPP, iii) digital mock-up, iv) ergonomics, v) robot simulation, vi) virtual commissioning, vii) life-cycle assessment, viii) layout planning, and ix) material flow simulation. The following sections provide a brief description of the use of simulation for each application.

The first and most important modelling tool that is considered is the Computer-Aided Design (CAD) software. It is used for two-dimensional and three-dimensional engineering drawings of products and plays an important role in product, equipment and system design. The various design concepts are visualised during this stage and it is predominantly used in the planning phase. It is generally used for geometric modeling, engineering analysis, simulation and scientific computing [Xue, 2018].

The next application is the Computer Aided Machining that is usually prevalent during the later stages of the planning phase as shown in [Mourtzis, Doukas and Bernidaki, 2014]. With the help of computer-aided machining, it is possible to convert the CAD design that was created during the planning phase into a set of manufacturing instructions. This is achieved using G-code, which is a language used by numerically controlled machine tools. CAM reduces the involvement of human operators, enables automation of the manufacturing process and subsequently reduces cost and increases profit [Xue, 2018].

The next application that is considered is the analysis of multiple physical phenomena such as heat transfer, acoustics, fluid flow, etc. using multi-physics modelling. Multi-physics modelling is generally used in a number of applications in the manufacturing industry to analyse the stress distribution, thermal performance and various other structural and functional analyses. In the system lifecycle, it is generally located after the CAD modelling since the created CAD models play an important role in the analysis [Zhang and Cen, 2015].

The next application that is considered is the ergonomic analysis. Ergonomics deals with fitting the tasks and processes to the operator. It comprises of various investigations regarding the musculoskeletal, psychophysiological, cognitive aspects, etc. In general, this is predominantly done before the operational phase to ensure that the operators have good working conditions that does not hinder their efficiency [Stanton et al., 2004].

Robotic simulation includes the use of software that specifically model the robot behaviour which allows the analysis of kinematic, robot interaction, collision detection and path or motion planning [Rohmer et al., 2013]. This allows the planning for automatic and semi-automatic operations. In a similar way, various software tools such as Delmia Ergonomics and Tecnomatix Jack allow the modelling of the human operators and the tasks they perform during the planning phase. The impact of various factors such as noise level, humidity, shift pattern, job rotation, work teams, hierarchy, diversity, age, gender, skill level, etc., on the performance of the operator is analysed [Baines et al., 2005]; they are generally done during the implementation and operation phase.

Supply chain modelling is important during the implementation and operational phase of the lifecycle. A supply chain network comprises of the material and information flow and it is necessary to understand where in the supply chain the risks might occur. It also supports operational policy selection and decision making [Gjerdrum et al., 2001]. By effectively modelling the supply chain using software such as Simio, AnyLogic, etc., the costs associated with the inventory, warehouse, logistics and potential risks can be monitored and controlled.

Layout planning involves the allocation of the resources such as machines, operators, robots, vehicles, etc., within the defined space. It also deals with the grouping of similar machines such that the efficiency of the operations can be increased. It is also important to consider the resource orientation, space consumption, resource location constraints and collision possibilities [Jiang et al., 2014] and software such as SmartDraw and Delmia Plant Layout Designer can be used for this purpose. The layout planning needs to be considered every time a scale-up project happens.

Computer Aided Process Planning (CAPP) deals with deciding the method of manufacture or assembly of the product by using software. It is a very important activity that is typically done once the product designs are finalised. It also comprises of determining the sequence of operations and optimisation of the process parameters. Using CAPP, it is possible to automate the process planning activities [Trstenjak and Cosic, 2017].

Material flow activities are planned in parallel to the layout planning activities. Material flow simulations allow the comparison of alternate scenarios in a stochastic environment [Reinhardt et al., 2019]. They also allow planning for the quantity and path that these resources need to take. It is also important to check for any potential risks and hazards that the use of AGVs or vehicles might have on the manufacturing system. Additionally, the collision detection, efficiency and travel time can also be investigated using material flow simulations using software such as Visual Components, Ciros, etc.

Other applications include Augmented Reality (AR) and Virtual Reality (VR) which enable the digitalisation of manufacturing system. They are increasingly used to support system modelling. The interest towards these technologies is on the rise [Bottani and Vignali, 2019].

The modelling of maintenance involves the continuous monitoring of the system and its resources. It aids the decision making regarding maintenance activities and reduces the risk of equipment failure [Okoh et al., 2017]. The maintenance decisions also affect the operational performance of the production system. Therefore, enough emphasis needs to be given to the modelling of maintenance activities; it is possible to perform analysis regarding maintenance using DES software such as Lanner Witness, FlexSim, etc.

2.6.4 Simulation method selection

Although there are different methods and techniques of simulation, within the scope of this research, the DES and kinematic modelling software are considered. The reasons for selecting them are discussed below. There is need to use a simulation method that has the capability to model workstation, operator, material handling, process and other manufacturing resources, in addition to having the capability to calculate the station processing time. For this purpose, the kinematic modelling tool is considered to be the most suitable candidate. The second simulation method should be capable of modelling the assembly line, material handling and product flow, in addition to having the capability to perform operational research. From a review of literature, the three simulation techniques that are chiefly used for manufacturing system modelling are DES, Agent-based Modelling, and System Dynamics (SD). Their suitability for DDSM will be discussed in the following paragraphs.

SD is a continuous simulation technique that has two main aspects: qualitative and quantitative. Causal loops are drawn to show the way the elements of system are related and SD models are, in general, not used for optimisation or prediction. Commercial SD software include Vensim, Stella, iThink, Powersim, and Simile. SD methods are widely used tools for logistics and supply chain decision making [Tako and Robinson, 2012]. SD models are typically modelled at a higher level of abstraction than DES [Brailsford et al., 2014]. Moreover, SD is a deterministic simulation approach and hence not suitable to model the randomness that is a characteristic of the scale-up phase as previously explained. Also, the level of detail in the SD simulation is not enough to model a detailed assembly line. Therefore, SD is considered unsuitable for DDSM.

Another prevalent simulation tool is the Agent-based Modelling, where a complex system is modelled as autonomous agents and rules describe the interaction between agents. This helps to see various patterns that were not initially programmed to emerge [Macal and North, 2010]. Agent-based modelling is typically used to model war scenarios, spread of epidemic, supply chains and stock markets. Agents get information from environment and take decisions based on rules. Agent-based modelling is a stochastic technique and a bottom-up approach [Maidstone, 2012]. Agent-based simulation platforms include MASON, Repast (JAVA and Objective-C versions), Swarm, and NetLogo [Seleim et al., 2012]. Agent-based modelling is not as established as DES for research on queuing systems and the models are more difficult to develop, especially during the planning stages [Maidstone, 2012]; hence, the Agent-based modelling technique is not considered for the purpose of this study.

The last simulation method that is considered for the scale-up modelling is the DES since it is an established technique for analysis in systems where queuing is evident. DES also models the manufacturing system at a higher level of detail which is important for simulating the workstation details. Moreover, DES is relevant for modelling stochastic scenarios and produces different results each time the simulation is run due to the use of probability distributions. Additionally, it is suitable for modelling the manufacturing system in the concept phase to compare different *'what-if'* scenarios and system designs [Maidstone, 2012]. A comparison of the DES, Agent-based and SD methods is provided in **Table 2.3**.

Table 2.3: Comparison of modelling using DES, Agent-based and SD methods.

Subject	DES	Agent-based	System Dynamics
Level of detail	High	High	Low
Level of abstraction	Low	Low	High
Suitability for queuing system	High	Low	Low
Modelling randomness	Yes	Yes	No

Across the representation of simulation tool usage in **Figure 2.10**, DES can be used during various phases depending on the application ranging from layout planning, resource allocation to process planning. Similarly, kinematic model can be used across the lifecycle for process planning, robot and operator modelling, path planning, collision detection and such. This underscores the importance and need for simulation and modelling tools for product lifecycle management and decision making activities.

In this research, it is hypothesised that the data integration of DES software and kinematic modelling tool will support decision making during scale-up planning phase. Hence the next few sections will touch upon the workflow and functioning of DES and kinematic modelling software.

2.6.5 Simulation using kinematic modelling software

The family of software that enable the virtual building and testing of manufacturing workstations prior to the commissioning of the production line are cited using many related terms such as 'Virtual Engineering' [Ghani, 2013],'3D Simulation' [Wischnewski et al., 2012; Caggiano, 2010],'Digital Mock-up' [Mourtzis, Doukas and Bernidaki, 2014], etc. To achieve consistency in the use of words related to the aforementioned family of software, the author intends to use the term 'kinematic model' due to their inherent capability to model the kinematics and motions of various workstation elements. Using kinematic modelling software, digital models are created to simulate the production system layout, process plans and system configurations. They also allow the verification of assembly processes in the absence of a physical system [Maropoulos and Ceglarek, 2010]. They can be used to represent how the real workstations would look like on completion [Ghani et al., 2015].

Primarily, a kinematic modelling software consists of three dimensions: kinematics, behaviour and reference co-ordinates. The kinematics dimension encompasses the information related to the geometry, links and joints, whereas, the behaviour dimension comprises of transition and states [Liu et al., 2012]. Moreover, collision detection, robot simulation and material handling system selection can be performed with such simulation techniques [Caggiano and Teti, 2012]. According to Caggiano [Caggiano and Teti, 2018], the input data such as digital human models, kinematics and geometry of robot, geometry of equipment and cell layout are necessary for creation of a model. The output data that could be received from the simulation tools include the collision free paths for robots, operators and AGVs, and optimal station layouts. The use of kinematic modelling to virtually engineer and validate the workstations, robots, and other resources is anticipated to reduce the ramp-up time [Falkman et al., 2009]; the building and testing of a system virtually provides cost and time savings. For example, Seidel reports that the modelling of material handling solutions helped reduce the commissioning time by 25% in an industrial facility [Seidel et al., 2012] and Makris highlights that the use of virtual commissioning helped reduce cost by 15% reduction of human resources [Makris et al., 2012]. Typically, the workflow of a kinematic model generation involves the following steps: CAD processing to remove unnecessary geometry, export of CAD models to either Virtual Reality Modelling Language (VRML) or Jupiter Tessellation (JT format), assembly of components within the kinematic modelling tool, modelling the kinematics, connection to physical system and validation of the models [Bathelt et al., 2005].

Depending on the software that is used, the CAD processing activity can be performed at varying levels of detail that enable the selection of identical elements, replacing selections, closing gaps in the CAD surface, deleting geometry, closing drill holes, reducing polygons, simplifying round edges and merging vertices. To simplify the process of gathering layout data, some software even have point cloud support to automate the layout capture. The exact procedure of data capture, CAD processing, kinematic definition, logic building, etc., varies depending on the software used but the set of data that are required for the kinematic modelling software generally remain the same across the tools. This will be discussed in greater detail in the next chapter.

2.6.6 Simulation using DES software

A typical DES software comprises of a system clock, event list, statistical counters, system date and report generator that enables it to record the activities and statistics of a simulation model. In DES, unlike continuous simulation, the focus is on the change of system state which is referred to as an *'event'*. The events happen at irregular intervals and hence the system clock jumps to these discrete points of time when the state changes occur. This makes the DES model efficient and fast [Brailsford et al., 2014].

A DES software generally comprises of certain characteristics that enables it to perform the various analyses. They are as follows: i) a DES software comprises of a number of statistical distributions for modelling the shop-floor data, ii) it also uses pseudo-random numbers that enable the modelling of stochastic systems, iii) it also has the capability to replicate the simulation runs with different streams of random numbers, and iv) it has the capability to provide results in confidence interval and can take an input value for warm up period to achieve better statistical results by waiting for the system to reach steady state. [Chryssolouris, 2013].

A literature review performed by Jahangirian [Jahangirian et al., 2010] and [Mourtzis, 2020*b*] about simulation techniques indicated that the use of DES in manufacturing had considerably increased throughout the years. DES is a very promising tool for manufacturing systems and helps experiment different strategies and system configurations for decision making [Negahban and Smith, 2014]. Manufacturing systems have complex interdependencies that prove difficult to be contemplated by the human brain; this gap can be filled by use of DES for decision support [Barlas et al., 2014]. Although DES was initially created for manufacturing systems, it later gained popularity in other fields such as healthcare, transportation, etc. [Robinson and Brailsford, 2014]. In DES, elements with attributes are modelled to perform activities and it is primarily used in two major phases of manufacturing system lifecycle: design phase and operation phase. During system design phase, the applications involve layout planning, material handling design, manufacturing line design, etc. During operational phase, the applications involve operation planning, real-time control, scheduling, etc. DES can be used for comparing different scenarios for decision making and can help predict operational performance and utilisation of production lines [Azab et al., 2012]. The benefits of DES during the planning stages to enable fast decision making and reducing time-to-volume are also highlighted by [Kampker et al., 2017]. In their work, they have presented a methodology to use DES to support the early developmental phase and modelling of scalable production system.

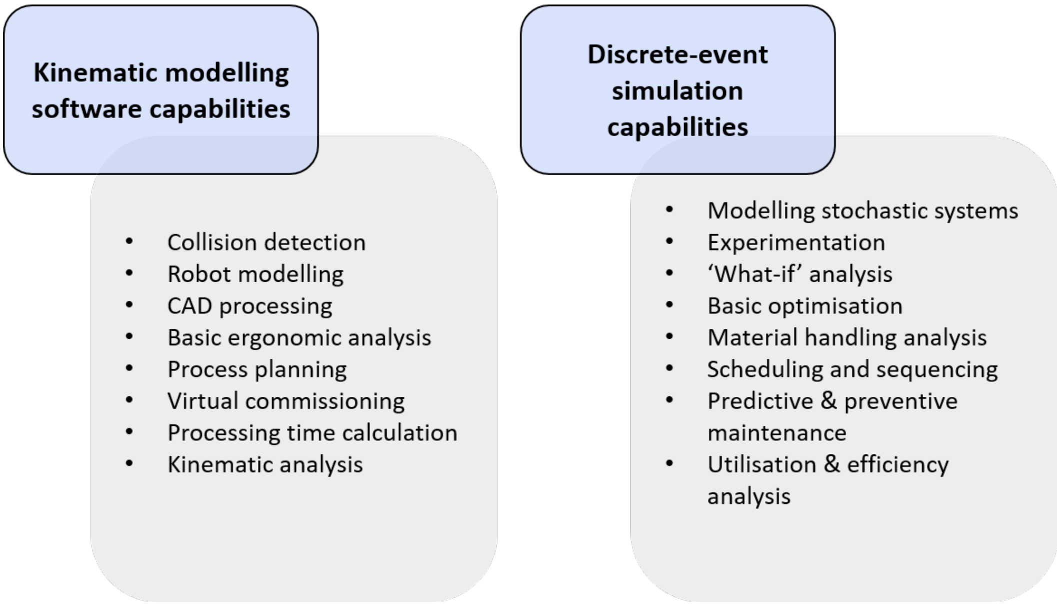


Figure 2.11: Capabilities of DES and kinematic modelling software.

This summarises the details pertaining to the kinematic modelling software and DES and **Figure 2.11** highlights the capabilities of both software. The following sections introduce the various standards that are relevant to the integration of simulation models.

2.7 Integration of simulation models

A multitude of simulation and modelling tools are currently employed in various manufacturing industries at varying levels of detail. The increasingly complex manufacturing systems need the assistance of different types of engineering tools for various reasons ranging from data storage to decision making. In addition to ensuring that these models communicate among themselves, it is also important to ensure that they are properly integrated with the data sources from physical system. It is important to ensure seamless communication between simulation models and other elements of the manufacturing systems. This also supports the re-usability of the software tools. However, not all engineering tools are developed in a standard way that provides a platform for communication.

As previously mentioned in section 2.3, PLM tools represent the family of software suites that provide functionalities that are intended to support the whole manufacturing lifecycle. Although they are very beneficial and support the manufacturing system in several dimensions, the software tools have their own proprietary format that makes it difficult to exchange data. Together with the lack of transparency about standards, protocols and technology, the key intention of such commercial tool developers is not in the best interest of the user [Popplewell et al., 2010]. Such problems exist in the domain of a homogeneous set of tools that are integrated over a common platform, but similar problems also hinder the interoperability across the wide spectrum of heterogeneous tools, each specialising in a specific domain. Terkaj and Urgo [Terkaj and Urgo, 2015] argue that a lack of effective interoperability among the multitude of software, reduces the support provided by them for complex manufacturing systems. Two possible ways to overcome the aforementioned issues are co-simulation and the use of neutral data format for exchange of information across various tools; this could reduce the cost of simulation model building [Shao et al., 2010; Liu et al., 2014]. The DDSM framework will comprise of more than one software to support the scale-up phase of the lifecycle and hence it is important to consider how the software will communicate with each other. Several works are available in literature in the areas of communication, common software platform, data exchange and interoperability. A few relevant ones will be discussed in this section.

2.7.1 Existing frameworks

Tolio identifies the characteristics that a virtual factory platform should have as i) the ability to handle heterogeneous information from various phases of lifecycle, ii) integration of knowledge and information from various tools at varying levels of hierarchy, and iii) the ability to allow engineers to use and exploit the advantages of simulation without the experience of specialists. Most importantly, their work highlights the need for a shared platform where data can be accessed and input by different tools [Tolio et al., 2013]. Although a number of standards and frameworks are proposed for the common data model, they are specific to certain areas of the factory. It is highlighted that there are three major approaches to achieve the data exchange between data management systems and software i) ontology and semantic web technologies [Terkaj and Urgo, 2015], ii) Standards, and iii) Application Programming Interface (API) and web services. In this regard, the XML (eXtensible Markup Language) is found to be the most used standard language [Penciuc et al., 2014].

The various prominent work in the area of software interoperability for data exchange are highlighted below. A neutral data format, **Automation ML**, based on XML for data communication between engineering tools was developed with the collaboration of Daimler, ABB, Siemens, Rockwell, Kuka, Zühlke, netAllied as well as the Universities of Magdeburg and Karlsruhe [Hundt et al., 2008]. The goal of AutomationML is to interconnect heterogeneous automation tools. Accordingly, typical objects in plant automation are classified as topology, geometry, kinematics and logic (sequencing, behaviour, control). The inherent architecture in AutomationML is distributed; the core concept being the top level data format using (Computer Aided Engineering Exchange) CAEX, geometry and kinematic storage using (COLLAborative Design Activity) COLLADA and logic information using PLCopen. Using these existing formats, AutomationML defines the association between them to favour data communication [Hundt et al., 2008].

Another architecture is the **CIMOSA** (Open System Architecture for Computer Integrated Manufacturing) proposed by ESPRIT consortium AMICE, which is a reference architecture for enterprise modelling to exploit the enterprise knowledge. It is intended to support decision making in production systems throughout the lifecycle [ESPRIT Consortium AMICE Staff, 1993] and monitor model-driven operation systems. In CIMOSA, four different modelling views are proposed and they are function, information, resource and organisation; it follows enterprise engineering concept and decouples the functionality and behaviour; the concept is validated in a number of case studies where the identified benefits include increasing efficiency of modelling and decision support systems [ESPRIT Consortium AMICE Staff, 1993].

Another relevant framework is the **Factory Data Model** (FDM), which is essentially a blueprint of the information available in the enterprise which is stored in an object-oriented database; a single FDM can go on to support many factory models and it is based on the gradual collection and refinement of data. The model proposes how a manufacturing system should be designed and performance evaluation should be done by using a factory model and data warehouse [Harding and Yu, 1999].

[Bloomfield et al., 2012] proposed the **Core Manufacturing Simulation Data Information Model** (CMSDIM) to enhance interoperability between the manufacturing systems and software tools. A standard XML-based schema is adopted to standardize the transfer of information. The proposed standard is found suitable for providing manufacturing data to software used in the simulation environment. CMSDIM encompasses six packages: layout, part information,support, resource information, production operations, and production planning [Bloomfield et al., 2012].

[Terkaj et al., 2012] proposed the **Virtual Factory Data Model** (VFDM), built using semantic web technologies, the aim being to develop an integrated virtual environment that supports the data transfer between the software and factory throughout the lifecycle. The framework consists of the Virtual Factory Data Model, Semantic Virtual Factory Manager and Decoupled Virtual Factory Modules. It provides a data model to cover the following domains: building, product, process, resource, production system and factory. The VFDM forms one of the three pillars of the (Virtual Factory Framework) VFF European project. The framework was demonstrated with a test case using a non-commercial software and its ability to exchange data as aspired [Terkaj et al., 2012].

Briefly touching upon another relevant work, the **Knowledge Configuration Model** (KCM), is a knowledge management model based on the web services-based interoperability; one of the core issues it can help mitigate is the problem arising from the diversity of simulation software [Penciuc et al., 2014].

The **ISA 95** is a standard developed by Instrumentation System and Automation which is adopted under IEC/ISO 62246 to provide integration between control systems and enterprises [He et al., 2012; Harjunkoski and Bauer, 2014]. The **Integrated Simulation Method** (ISM) is another approach to support data management of digital factory which provides an architecture that comprises of six layers and a Virtual Factory Data Management System [Zhai et al., 2002].

A standard XML based neutral data format called PPRX to exchange data between heterogeneous PLM tools with the help of PLM integrator was proposed by Choi et al [Choi et al., 2010]. In their work, the data from the PLM tool is exported based on PPRX that represents the (Product Process Resource) PPR information.

This section provided an overview of the existing frameworks for digital factory integration and a summary is provided in **Table 2.4**; the next section delves deeper into the knowledge available on data integration of DES and kinematic modelling software.

2.7.2 Data integration of DES and kinematic modelling software

Digital factory represents a model that consists of various elements, automation assets and involves mapping their behaviour and relationships [BSI, 2016]. Within the wider concept of digital factory, there needs to be considerations about the data exchange, not only from manufacturing systems to modelling software, but also between the heterogeneous modelling software. Specifically, the data integration between DES and kinematic modelling software is considered beneficial [Caggiano and Teti, 2018, 2012]. Although DES can be used to perform simulation models that allow operational research, it lacks the underlying information that is necessary to identify the feasibility of the modelled scenarios with respect to the station

Table 2.4: Existing frameworks for integration of software and data management systems.

Framework	Approach	Key features/ Aim
AutomationML [Hundt et al., 2008; Drath et al., 2008]	XML-based data format	Distributed architecture
		CAEX as top-level data format
		COLLADA for geometry and kinematic storage
		PLCOpen for logic information
CIMOSA [ESPRIT Consortium AMICE Staff, 1993]	Reference architecture	Function, Information, Resource and Organisation modelling views
	for enterprise modelling	Aim to increase modelling efficiency
		follows enterprise engineering concept
FDM [Harding and Yu, 1999]	Object-oriented database	Defines how to design manufacturing systems
		using factory model and data warehouse
CMSDIM [Bloomfield et al., 2012]	XML-based format	Enhancing interoperability between system and software
		Consists of six packages - layout, part information,
		support, resource information, production operations & production planning
VFDM [Terkaj et al., 2012]	semantic web technology	Data model for product, process,
		resource, production system & factory
KCM [Penciuc et al., 2014]	web services-based	Knowledge management model
	interoperability	Helps mitigate problems arising with simulation software diversity
ISA 95 [He et al., 2012; Harjunkoski and Bauer, 2014]	Standard (IEC/ISO 62246)	Integration between control systems & enterprises
ISM [Zhai et al., 2002]	Data management system	Supports data management of digital factories
PPRX [Choi et al., 2010]	XML-based format	Exchange of data between heterogeneous PLM tools
		using PLM integrator

processing times, machine failure, robot and digital human operating times, etc. Especially in the early planning stages, the exploitation of the capability of DES to perform statistical analysis is restricted due to the lack of accurate data [Ghani, 2013]. During the concept and planning stages, for entering the time values within DES, the target cycle time or process time is generally used instead of the actual production time. However, the accuracy of input data in DES is very important for meaningful results [Mieth et al., 2019]. In such situations, information from the kinematic models that are used to analyse the workstations can complement the existing capabilities of DES models [Caggiano and Teti, 2018].

Ghani et al. [Ghani et al., 2015], have in their research studies proposed approaches to perform this integration between kinematic model and DES. In order for successful integration of the two software, it is essential to understand the common data that is required by both simulation models. The integration also paves way for understanding the impact of changes in workstation level on the system behaviour at the higher level which enables better decision making. In this regard, it is worthwhile to spend time to understand about the format of data exchange to make it generic for the integration of any kinematic model to DES software. In the algorithm proposed by Ghani for the integration, process time and maintenance data are passed to DES using a middleware for data processing with the XML format. It is also identified that there is lack of data compatibility between DES and kinematic model software. From Ghani's thesis [Ghani, 2013], the possibility to encapsulate parametric data such as velocity, torque, motion time of the various considered elements within the kinematic environment is explored in detail. This enables the reuse of data which ultimately aligns with the goal of improving accuracy of DES models for early lifecycle production systems that are yet to be commissioned. In another related work, the integration of kinematic model and DES for visualisation purpose is discussed; the need for a neutral format for data exchange is also emphasised [Kibira and McLean, 2002].

In order to understand the best means of data exchange between the DES and kinematic modelling software, it is important to understand the various types of integration. Since the DES models represent the assembly system and the kinematic models represent the workstation levels, this integrated model is considered as a multi-scale model. Multi-scale model is associated with the modelling of systems at different levels of abstraction [Brailsford et al., 2014]. Single scale models that are independently validated are integrated to form the multi-scale model and the integration can be one of the following five types.

- Serial method: A single model on one scale passes information to a model in another scale and they operate sequentially.
- Simultaneous method: Models operate simultaneously; lower scale models are used to generate system information which is sampled and used by higher scale models.
- Hierarchical method: Lower scale models are embedded within higher scale models and they operate simultaneously.
- Multi-domain method: Information between lower scale and higher scale model is passed using an interface.
- Parallel method: Model comprises of several multi-scale models.

From the considered integration types, the multi-domain method is the most suitable one for the considered scenario. The data from the lower level workstation models are intended to be passed to the higher level assembly system models using an interface. In the case of DDSM, this interface is a database. This multi-domain method of integration can be employed for supporting the decision making process. The simulation-based decision making forms the major part of the DDSM framework and a detailed explanation of the methodology will be provided in the next chapter.

2.8 Critical analysis of relevant work

2.8.1 The aspect of scalability

A plethora of papers discuss scalability of manufacturing systems with most of the papers focussing on reconfigurable manufacturing systems. From a survey of related papers, two important principles for implementing scalability that can be subsequently adapted for scale-up are identified as i) linking or adding identical elements/stations to increase the productivity, and ii) increasing / decreasing the performance of an element/station by changing its functionality [Putnik et al., 2013; Fricke and Schulz, 2005]. Principle one is referred to as resource replication and principle two is referred to as resource upgrade. The replication principle is also referred as parallelism, however, unrestricted replication is not possible because of the space and budget limitations [Putnik et al., 2013].

In this regard, a method of quantification of scalability within the wider context of changeability is proposed by [Ross et al., 2008]. The relationship between changeability and scalability is explored in their work. A notable work that discusses cost modelling for capacity scalability is proposed by [Deif and ElMaraghy, 2007*b*]. From their research, Reconfigurable Manufacturing Systems (RMS) are inherently scalable systems and are modular enough to be scaled-up in both physical and logical domains. The approach generates an optimal capacity scalability schedule with the help of GA, which highlights when, where and by how much system should be scaled-up to meet the new demand. In another related paper, [Deif and ElMaraghy, 2007*a*] have discussed the importance of managing capacity scalability and have assessed alternate strategies for different demand scenarios for RMS with the help of a System Dynamics model. The core aspect of both reviewed research works is fixed upon the capacity scalability of RMS; the considered scale and frequency of demand change are significantly different to the scale-up phase considered for DDSM.

Considering scalability planning and management at the production line level, [Almgren, 2000] emphasises the importance of identifying disturbances during pilot phase and classifies them as internal and external. Internal disturbances include setup time, breakdown stoppages, operator performance and motivation; external disturbances emerge from material quality issues. The understanding of the disturbances can help smooth the transition from the pilot phase to operational phase. Consequently, the importance of modelling failure, breakdown of workstations and introducing randomness is elicited clearly in their work. Wang and Koren have presented a GA-based optimisation algorithm that can help decision making regarding adding or removing machines from production lines in the event of new market demand [Wang and Koren, 2012; Koren et al., 2017]. Although the approach considers task reallocation and specifies the quantity and location of where the new machines can be added, the material flow, labour, operational cost and space occupancy are not considered in much detail. Moreover, the workstation level configuration changes and production line scheduling and dispatching rules, which have an impact on the scale-up, are not discussed comprehensively.

2.8.2 Manufacturing system and station configuration selection

There are a number of papers tackling the problem of assembly system configuration and line balancing. The presented discussion reviews only a few among the several existing publications on assembly system configuration selection; specifically, those that are relevant to the DDSM framework are selected for discussion.

In the domain of workstation level scale-up, [Bensmaine et al., 2013] have tackled the machine selection problem for RMS with Non-dominated Sorting Genetic Algorithm (NSGA - II). The main focus of their work is the selection of machines from a candidate list considering two main criteria, the minimum total cost and minimum total time. In order to achieve this, the multi-objective optimisation method called NSGA-II was employed with the objective of minimising the time and cost while selecting a set of machines that can perform all the required operations. The cost function in their problem formulation comprises of four elements: machine usage cost, configuration change cost, tool change cost and tool usage cost. Similarly, the time function comprises of the processing time, tool changeover time and configuration change time. Their methodology is demonstrated with a case study that comprises of ten candidate machines and a product comprising of three features that need certain operations. By using their method, the decision makers can select the suitable machines that would fulfill their requirements. Although the approach is very beneficial and supports decision making, it is primarily focussed on RMS, Reconfigurable Machine Tools (RMT) and machining operations. Additionally, the initial selection of candidate machines is subjective and experience-based; discussions relevant to the method for calculation and sourcing of process time data is lacking. Moreover, product variants are not considered in their approach.

[Manzini et al., 2018] proposed an integrated design approach in which four computational tools, namely assembly system configuration tool, assembly cell configuration tool, production planning and simulation and reconfiguration planning tool operate using a database designed around the Core Manufacturing Simulation Data (CMSD) standard to support production system design and reconfiguration. To improve the design process, these tools also have feedback loop. Through the use of the tools, the following activities are carried out. Firstly, the system design that defines the number of cells, task allocation, process sequence and product routing is identified. Secondly, the detailed design of the assembly cell along with task sequence and allocation is identified. Thirdly, the capability of the selected configuration to meet the needs of the OEM are verified. Finally, the reconfiguration aspects of the selected design are considered. An illustration of their framework can be seen from **Figure 2.12**. In their research, the general template of an assembly cell comprises of a seven-axis robot for handling and transportation with stations around a central rail.

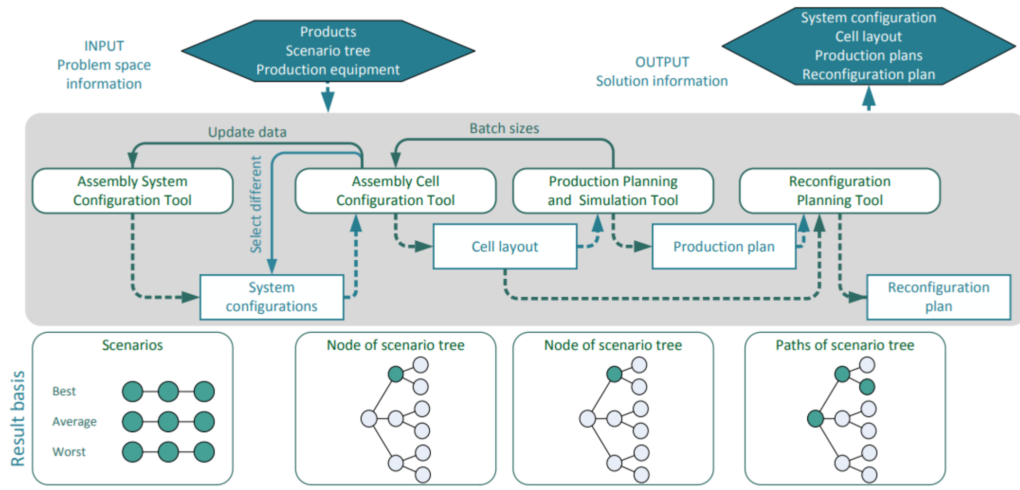


Figure 2.12: Integrated design approach proposed by [Manzini et al., 2018].

The key focus of their work lies on the automated design of production systems and selection of suitable assembly system configurations, detailed analysis of the physical layout, task sequencing and dynamic performance evaluation, predicting future scenarios and operation-related cost, and guiding the reconfiguration of assembly systems. The approach starts with the selection of production line configuration and drills deeper to define the workstation configurations for assembly systems at the early planning phases of the system lifecycle. The specific focus is provided to cellular architectures and modular assembly systems and the approach presents a top-down workflow which starts with the generation of high-level system configurations.

In summary, the approach provides a very detailed holistic approach to the assembly system design selection problem, however, the selection and decision making of system level resource architecture and layout before detailing the cell level leads to the analysis performed at a high level (assembly system) without data from the lower level (assembly cell). Though good configurations might be obtained, it is more beneficial to adopt a bottom-up approach where the data from the lower levels are encapsulated into the higher level model thus enabling more meaningful analysis without the conventional black box approaches. Additionally, details pertaining to the data architecture and workflow of integration is lacking.

[Guschinskaya et al., 2008] proposed a heuristics-based optimisation procedure for designing serial machining lines. Their objective is to minimise the machining line cost; constraints such as precedence, inclusion, station exclusion and block exclusion are considered in their optimisation model. Their formulation of the objective function is purely mathematical and their research work only considers a single objective. However, manufacturing systems have inherent randomness that is not considered as part of this research. Moreover, the consideration of a single objective is less beneficial than considering a manufacturing system problem as a multi-objective one. This is because the single-objective optimisation problem provides one solution that satisfies the considered objective, however, more than one objective needs to be considered to model the manufacturing system realistically.

A two stage approach for assembly line configuration selection and resource planning based on multiple criteria is presented by [Michalos et al., 2015]. Their approach is represented in **Figure 2.13**. The various criteria considered for configuration selection include, but are not limited to, the machine utilisation, energy consumption, investment cost, etc. The first stage comprises of an analytical calculation of the required number of resources and stations and the second stage in-

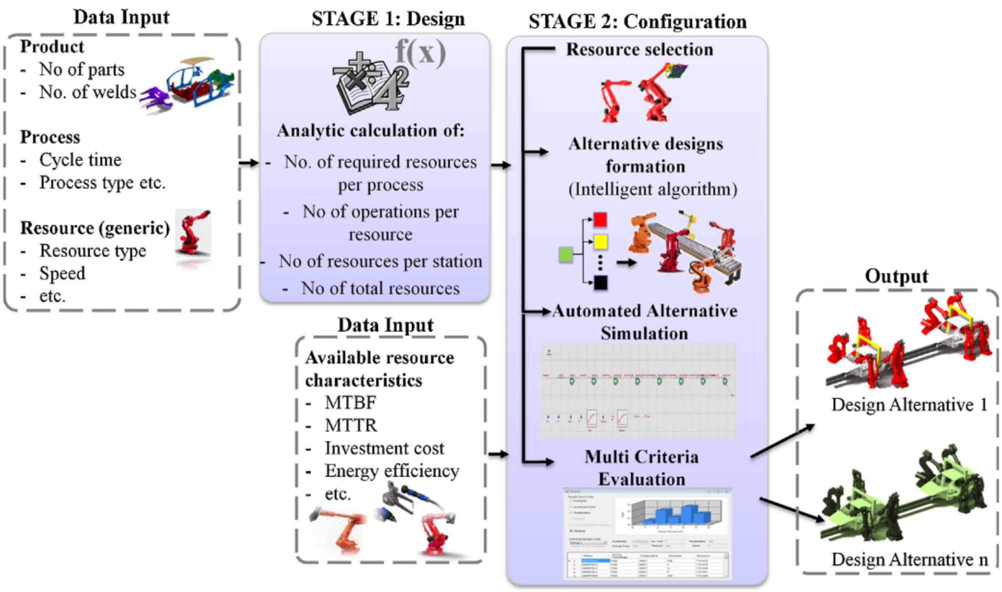


Figure 2.13: Assembly line configuration selection proposed by [Michalos et al., 2015].

volves DES-based assembly line configuration selection coupled with an intelligent search algorithm. The approach also benefits from automated DES model configuration. Despite the holistic view of the approach for assembly line design selection, the consideration of product variant and material handling units and the concepts related to interoperability and seamless data transfer among the software associated with design selection procedure are lacking. The major shortcoming has been identified as the time consuming calculation of process time; there is potential to overcome this issue with the use of kinematic modelling for comparison of design alternatives at the workstation level.

In the domain of system configuration selection for scale-up planning of Reconfigurable Manufacturing Systems, [Wang and Koren, 2012] have proposed a GA based optimisation procedure. The objective function is a mathematical formulation that seeks to identify the minimum number of machines required to meet the new market demand. The task allocation and machine allocation are the considered decision variables. The outlook of the approach is centered around a deterministic viewpoint of the system; the decision making process can also be made more flexible with the use of multi-objective optimisation to allow the comparison of alternate solutions.

Moreover, workstation configuration selection is not considered within the scope of their research.

[Ghani, 2013] has presented a framework using kinematic modelling and DES. In Ghani's research, kinematic modelling is used for visualising the reconfigurability, cycle time analysis and locating equipment; DES is used for analysis of productivity indicators. The integration of DES and kinematic model is actually beneficial because when using DES in the planning phase, lot of assumptions are made. In Ghani's approach, the process time analysis that is done in vueOne is used increase the accuracy of the DES models. The key application areas and focus of the work is on energy consumption, breakdown and human performance modelling at the kinematic model level. The method of integration proposed in Ghani's thesis is using XML data format; the (Virtual Driven Discrete Event Simulation) VDSim integration environment accepts input data, for instance, regarding maintenance and breakdown and the data processing is done using Visual Basic. This data is then sent to Excel and ultimately transferred to DES. Ghani has emphasized the need for structured information and the data model is specific for the application of reconfiguration and new product introduction.

Although their work emphasized the need for integration of kinematic modelling and DES within the context of virtual factory and proposed a novel approach for the implementation of such an integration, the work does not address the decision support required for scale-up planning and their data model is specific for their application. From the reviewed literature, the author finds that Ghani's work is the closest to what the DDSM framework intends to achieve. Therefore, the work done by [Ghani, 2013] will be explored in more detail and extended to provide a holistic approach that enables the workstation design selection, assembly system design optimisation and decision support for the transition to high volume.

2.9 Summary

The chapter started with an introduction to manufacturing systems and paradigms and funnelled towards the key concepts and ideas surrounding the term scale-up. To achieve consensus in the use of the term *'scale-up'*, it was defined and differentiated

Table 2.5: Summary of existing work related to scale-up.

S.No	Author(s)	Year	Principle management	Production line scale-up	Station scale-up	Scalability quantification	Capacity scalability
1	[Almgren, 2000]	2000	■				
2	[Fricke and Schulz, 2005]	2005	■				
3	[Deif and ElMaraghy, 2007*a*]	2007					■
4	[Deif and ElMaraghy, 2007*b*]	2007					■
5	[Ross et al., 2008]	2008				■	
6	[Wang and Koren, 2012]	2012		■			
7	[Putnik et al., 2013]	2013	■				
8	[Bensmaine et al., 2013]	2013			■		
9	[Koren et al., 2017]	2017	■	■			
10	[Manzini et al., 2018]	2018		■	■		
11	[Ghani, 2013]	2013		■	■		
12	[Michalos et al., 2015]	2015		■	■		
13	[Guschinskaya et al., 2008]	2008		■			

from related terms. The existing knowledge on scale-up were gathered from several academic articles and industrial white papers and critically reviewed to identify the shortcomings in planning and realising the scale-up phase. A significant proportion of the reviewed papers focus on the disturbances and events that occur during production scale-up and the approaches to manage and prevent these disturbances.

As perceived from **Table 2.5**, the body of literature holds relevant research works on scale-up. The majority of the studies focus on a particular aspect of the manufacturing system or have limited applicability. For instance, a detailed discussion might be provided for configuration selection of an assembly station but little or no discussion made for the assembly line configuration which is at a higher level of abstraction than the assembly station. There is lack of literature and research on an integrated approach for configuration selection that encompasses all aspects of a manufacturing system. Although some generic research work on assembly line design selection can be applied to the scale-up decision making problem, they do not tackle certain aspects of the scale-up problem. These primarily include i) application-specific criteria for decision making that are relevant for scale-up are not provided, and ii) scale-up relevant principles such as 'replication' and 'upgrade' are not considered. This makes it difficult to seamlessly incorporate the configuration selection methodologies to support the scale-up transition.

There is lack of a robust objective approach to select suitable production system configurations during the planning stages for scale-up implementation. Moreover, the industrial procedures for scale-up are identified to be ad-hoc, experience-based procedures that support the transition to higher production volume. This results in increased time-to-market, re-analysis or failure of scale-up projects due to poor system configuration choice, conducting expensive workshops and brainstorming sessions that rely solely on the opinion of experts to make decisions. There is also the risk of overstating the benefits of a system that leads to ignoring better solutions. If the chosen solutions are expensive in terms of capital and implementation cost and do not meet the required production capacity, it could lead to a situation where the cumulative system cost tends be relatively higher than those solutions that can cater to the required demand. However, manually searching the whole design space for complex manufacturing settings is not feasible. Therefore, the simulation-based scale-up planning activities were explored and the use of DES for performing what-if scenario analysis along with optimisation packages to explore the entire design space was suggested.

The use of DES as a stand alone model results in infeasible solutions which cannot be validated due to the absence of a physical model. To overcome this issue, this research proposes a bottom-up approach with parametric DES-based multi-objective simulation optimisation that obtains data from physical system and the kinematic model which increases the accuracy of the DES models even in the absence of a physical system.

2.9.1 Research gaps

The primary research gaps corroborated by the review are identified in this section. This step paves way for formulation of the research methodology.

Lack of a holistic approach

The review of current literature revealed that there are dissociated studies that relate to specific aspects of the manufacturing system design, configuration and scale-up implementation. These studies provide a deeper understanding of only a portion or abstraction level of the manufacturing system scale-up. However, they do not provide a holistic approach that considers the manufacturing system at varying levels of granularity. For example, the research done by [Bensmaine et al., 2013] focusses on the machine configuration selection, which is only one aspect of the manufacturing system. Their research does not detail the assembly line configuration selection following the selection of machines. Additionally, from a review of multiple approaches to assembly system design, [Manzini et al., 2018] have identified that there is lack of research on improving the higher abstraction level models, such as DES, based on data and solutions from lower abstraction levels, such as station level models.

System design selection for scale-up

Another important gap that was identified was the lack of design and decision support for scale-up [Terwiesch et al., 2001]. Although a number of papers discuss the scale-up management problem and explain the various strategies that can be considered for scale-up, there is not much work in scale-up specific assembly system design selection. The industrial white papers such as [Kinzoku, 2019; Volvo, 2007] also ascertain this fact by employing ad-hoc strategies to design the system for higher production volume. Therefore, the design space is not explored for better solutions.

Expensive solutions for scale-up management

Since the scale-up phase is part of the manufacturing system lifecycle, it is possible to apply existing PLM software to support the planning and decision-making activities. However, the commercial PLM solutions are expensive and cost around £100,000 and it is difficult to adopt them in small and medium size industries [Soto-Acosta et al., 2016; Terkaj et al., 2012]. If seamless communication could be established between heterogeneous software packages that perform the same tasks as that of the PLM, then this cost can be reduced. Therefore, it is necessary to identify a solution that can enable communication between heterogeneous software packages while also supporting scale-up decision making.

Need for data-driven approach

For scale-up management and decision making activities, it is important that the available data from the various sources are efficiently used. With the advent of Industry 4.0 and Internet of Things, it is even more critical that the proper sources of data for scale-up decision making are identified and embedded in the virtual models for scale-up planning. [Michalos et al., 2015] have identified the data that are required by the virtual models to perform assembly line configuration selection. However, this represents the generic data for assembly line configuration and whether this would be sufficient for scale-up planning is something that needs to be explored. Therefore, there needs to be more work done in this area to fulfill the gap that the lack of data-driven scale-up approaches have created.

Chapter 3

Methodology

3.1 Overview of the approach

The previous chapter highlighted the research gaps and underscored the need for a framework to support the scale-up phase, especially in the planning stages where critical decisions on assembly line modifications are done. To fulfill the research gaps and support the scale-up phase, a two-stage bottom-up approach named as the Data-Driven Scale-up Model (DDSM) is proposed and explained in this chapter. This two-staged approach aims to identify system designs that help realise scale-up by employing various digital manufacturing tools. The ultimate goal is to have a successful scale-up transition and to reach this goal, it is necessary to make modifications to the hardware and software. Essentially, this can be considered as an assembly line design configuration problem that is tailored for scale-up.

The DDSM is a bottom-up approach where the data from low level workstation models, built using kinematic modelling software, are leveraged to improve the accuracy and detail of high level system models in DES software to further perform meaningful analysis that support decision-making during scale-up phase. Since this type of behaviour modelling of complex systems demands a broad spectrum of software, data integration is all the more important. While commercially available software platforms for digital manufacturing promise interoperability among a multitude of software, their capability to support heterogeneous software is not quantified. The DDSM methodology does not consider software interoperability in much detail but touches upon certain aspects of it.

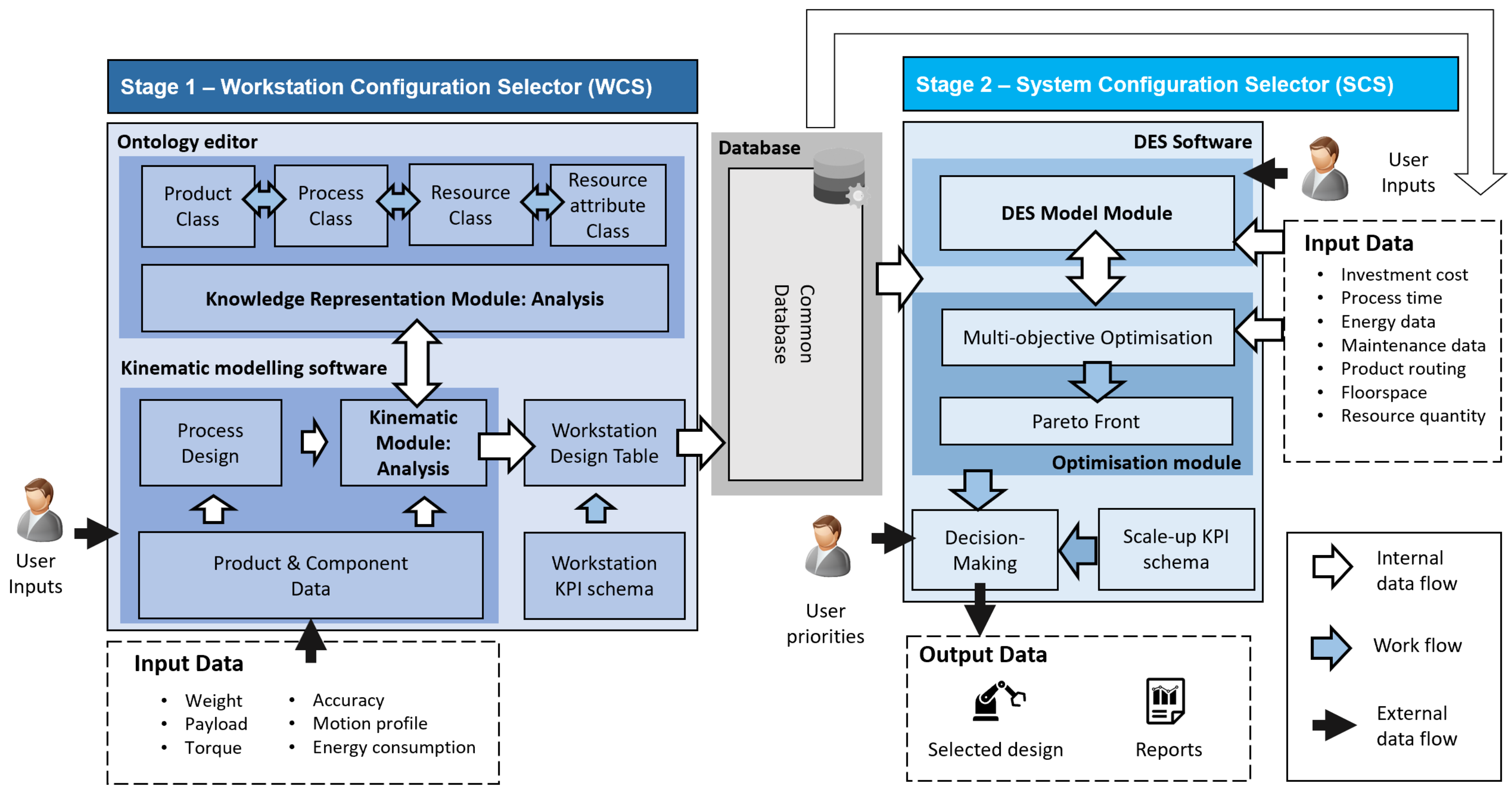


Figure 3.1: Schematic of Data-Driven Scale-up Model (DDSM).

The two main pillars of the methodology are the **Workstation Configuration Selector** (WCS) and **System Configuration Selector** (SCS) as represented in **Figure 3.1**. Stage one (WCS) is framed upon the assumption that the workstations comprise of one or more equipment to perform the required assembly operations on a product and that there can be different configurations or designs of workstations that perform similar, if not the same operations. It is to be noted that the term *'product'*, in the DDSM methodology, is used to represent the *'workpiece'* that will be assembled. From **Figure 3.1**, it can be seen that WCS comprises of two modules: the kinematic model module and the knowledge representation module. The details of knowledge representation module and the reason behind choosing ontology will be explained in section 3.3.2. Each module has different capabilities and they will be discussed in more detail in the later sections.

Stage two of the methodology comprises of two modules: the DES model module and the optimisation module. Again, the capabilities and method of data exchange between the modules will be explained in detail in the later sections. From **Figure 3.1**, the workstation KPI data from Stage one is accessed by the DES module for improving the accuracy of assembly line models. For example, the data such as the workstation process time, capital cost, operator allocation, etc., that are generally assumed within the DES model are instead obtained from Stage one. Ultimately, the data from the kinematic models at lower hierarchical level are accessible by the DES modules at the higher hierarchical level to subsequently improve the accuracy of input data. Furthermore, the DES model is coupled with the optimisation module, wherein, a multi-objective optimisation model for selecting near-optimal assembly system designs that minimise the scale-up cost and maximise the throughput is formulated. The steps involved in methodology can be summarised as follows:

1. Identify the current design of the assembly line that needs to be scaled-up and create virtual models of the assembly stations within kinematic modelling software. For this purpose, resource-related data and product-related data are necessary as indicated in **Figure 3.1**.

2. Within kinematic modelling software, identify process level details such as process sequence, task sequence and parameters for the assembly stations modelled and the product that needs to be assembled. This comprises of the

process design step in **Figure 3.1**.

3. The next step is to identify the set of equipment that can perform the required processes. To support the equipment selection process, query an existing equipment catalog or library within an ontology editor to retrieve those equipment that meet the process requirements. In **Figure 3.1**, this is indicated by the data flow between the kinematic modelling module and the knowledge representation module (ontology editor). Within the ontology editor, there are four classes that enable the equipment selection process.

4. Following the selection of suitable equipment, the resulting workstation configurations need to be validated within the kinematic model. The validated configurations are represented in the workstation design table along with their respective KPIs such as process time, energy consumption, cost, etc., and stored in the common database. More details about the database and workstation KPIs is provided in section 3.3.3.

5. Create a parametric DES model of the existing line and create communication interfaces with optimisation module and the database. This allows the exchange of optimisation variable values and workstation KPI data. This data is indicated as the input data to DES and optimisation module in **Figure 3.1**.

6. Code the fitness evaluation function/objective function and the optimisation algorithm within the optimisation module and set the initialising parameters. The data integration between the DES model module and optimisation module enables the simulation optimisation as indicated by the arrows in **Figure 3.1**.

7. Perform simulation optimisation, at the end of which the pareto front that represents the good solutions are obtained. The solutions represent the assembly line and station configurations; these solutions need to be validated in DES software.

8. The validated assembly line configurations are compared using Scale-up KPI schema and user priorities. The configurations are compared for decision making and displayed in a radar chart which indicates the assembly line performance with respect to a set of criteria.

Steps one to four provide an essence of how the WCS (Stage one) works. Steps five to eight provide details about SCS (Stage two). Section 3.2 provides details pertaining to the data modelling and investigation of the data required by kinematic modelling software and DES at various stages of the lifecycle and this is in alignment with research objective one. A thorough explanation of each stage of the methodology is provided in sections 3.3 and 3.4 of this chapter.

3.2 Data modelling along the system lifecycle

As explained in Chapter two, simulation and modelling play a vital role in different phases of manufacturing system lifecycle. Several heterogeneous software that specialise in computer aided design, computer aided process planning, augmented reality, digital-mock up, lifecycle assessment, ergonomics, computer aided manufacturing, layout planning, supply chain simulation, process simulation, etc. need to communicate with each other for successful modelling of complex manufacturing systems. Within the context of this thesis, the use of CAD, kinematic modelling, DES and their dependencies will be discussed in detail as they form part of the workflow. To better understand the dependency and allow data integration among them, it is crucial to understand when, where and how these software are utilised across the lifecycle and the type and frequency of data that are passed between the aforementioned heterogeneous software. **Figure 3.2** represents the use of DES and kinematic modelling software across the lifecycle of the manufacturing system which is explained in detail in the following sections.

3.2.1 Concept stage

From **Figure 3.2**, the manufacturing system lifecycle is represented on the *'y'* axis and includes the concept, implementation, operational and end of life phase. The arrows in the diagram indicate the direct or indirect data flow among the various modelling entities and between the modelling entities and assembly line . The *'x'* axis represents the assembly system, both physical and virtual; the former includes any tangible component in the real world manufacturing system such as fixtures, grippers, pilot line and assembly systems, and the latter includes virtual process and resource development and virtual product development. At the start of the manu-

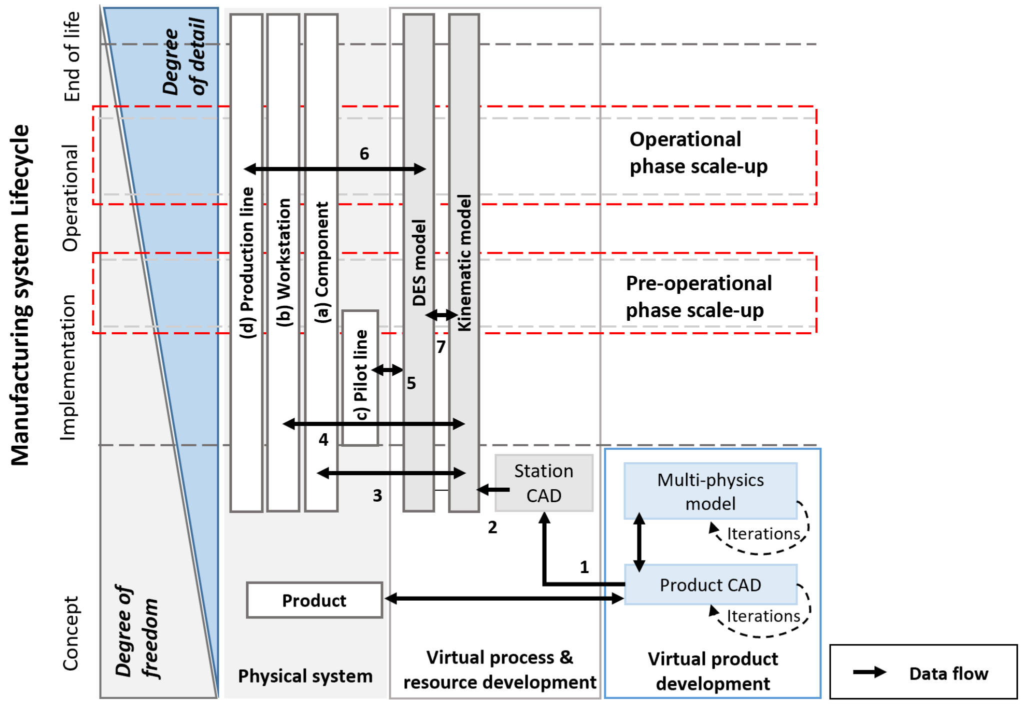


Figure 3.2: Use of DES and kinematic model software across the lifecycle.

facturing system lifecycle, in the virtual product development that happens during the concept phase, the data exchange between the CAD and multi-physics models helps analyse and validate the product design. This is represented in **Figure 3.2** using the arrow between product CAD and multi-physics model in the virtual product development dimension. Several iterations are done before the final product design is obtained. This acts as the catalyst for the virtual process and resource development where the station CAD models are manipulated to perform various analyses in the virtual environment. This is represented using arrow 1 between station CAD and product CAD. During the planning and development stages of assembly workstations, there might be situations where standard off-the-shelf equipment might be enough to perform the required assembly process. In some other situations, if there are no existing equipment that can perform the required assembly process, bespoke designs need to be developed. In case of such bespoke machine designs, initially, the design of machines are first created virtually followed by several iterations before the actual machine is accepted into the line. In case of standard off the shelf equipment, this process can be skipped and they can be readily modelled in the virtual environment using data obtained from data sheets and other such sources. This information can be encapsulated within kinematic modelling tool, as represented by the arrow 2 between the Station CAD and kinematic model in **Figure 3.2**, for planning the process. Moreover, the information regarding the product from the product CAD and relevant component data such as gripper range, robot payload, etc. from shop floor/physical system can also be accommodated within the kinematic model to enhance it. This process enables the creation of a library within the kinematic model that allows the re-use of the created virtual components for future projects.

3.2.2 Implementation stage

Progressing onto the implementation phase, the workstations can be commissioned after performing basic analyses such as collision detection and path planning within the kinematic model. The component and workstation entities in the real physical system, once they are commissioned and in operation, are potential sources of data. From **Figure 3.2**, the arrow between component (a) and kinematic model represented by 3 and the arrow between workstation (b) and kinematic model represented by 4 signify this. The pilot line (c) which is used to test the assembly of the product, comprises of the components (a) and workstations (b) and is also represented in **Figure 3.2**. The arrow 5 represents the data flow between the pilot line and the DES model. This data enables the modelling of pilot line within the DES software; the models can be refined and improved in an iterative manner. The pre-operational scale-up usually happens around this stage. When the pilot phase is completed and the line needs to be scaled-up, the existing workstation models of the pilot line within the kinematic modelling software can be used to further establish new configurations both at workstation and system levels. The assembly line designs can be modelled within DES which allows the system designers to make better decisions. This enables the conceptualisation of the production line even before its implementation. The arrow 6 represents the data flow from the production/assembly line to the DES models. The analyses performed within the kinematic modelling software, with proper integration, can provide useful data for the DES model and this is represented by arrow 7.

3.2.3 Operational phase

Moving on to the operational phase, there is opportunity to get more data from the assembly lines to update the virtual models such that they better represent the physical system. At this stage, data from the real system can be used to support the DES models to further analyse potential scenarios. The operational phase scale-up might happen due increase in product demand while the assembly line is operational. The connection between the DES model and the real system can be classified as on-line or off-line. In off-line simulation, the model is not coupled to the real system; in on-line simulation, the model is connected to the real system and the collected data can be used for real-time planning and scheduling [Mirdamadi et al., 2007]. Despite the benefits of using the data for improving potential DES models, it has been identified that they are typically not in a form which is readily usable by the simulation software. Another major issue during this phase is the possibility of human error when manually entering data [Mieth et al., 2019]. To get more clarity on this, the various types of data available and their usability is investigated in the following section. The end of life is not considered within the scope of the thesis and hence will not be discussed in detail.

3.2.4 Data usage across the system lifecycle

This section discusses the various data that are required at various stages of the lifecycle from both the shop floor and other simulation models. They can be classified into datasets as follows.

- **Dataset 1** - Data from the CAD model for creation of component and station level models; this data is further used in kinematic modelling tool during the concept phase and is represented by arrows 1 and 2 in **Figure 3.2**.
- **Dataset 2** - Data from the kinematic model of the workstations and components to the shop floor; this data is used to commission the new pilot line and is represented by arrows 3 and 4 in **Figure 3.2**.
- **Dataset 3** - Data from the real pilot line workstations and components to kinematic model and DES in the implementation phase; this data is the feedback from the physical system to improve the virtual assembly line, workstation

and component models. In the **Figure 3.2**, this dataset is represented by arrows 3, 4 and 5.

- **Dataset 4** - Data from kinematic model to DES to create the new scale-up line model; this represents the pre-operational scale-up and the workstation and component virtual models within the kinematic modelling software can be used to support the DES modelling during the implementation phase. This dataset is represented by arrow 7.

- **Dataset 5** - Data regarding the assembly system from the DES model to kinematic modelling software. The analysis performed within DES might indicate the lack of productivity and this is useful feedback to workstation models within kinematic modelling software. With this feedback, the workstation design can be modified by replacing the existing equipment with more productive ones. This dataset is again represented by arrow 7 and is used during implementation phase.

- **Dataset 6** - Data from the DES model to the assembly line or shop floor; this data is beneficial for commissioning the new pre-operational phase scale-up design during the implementation phases. Based on the analysis performed within the DES software, the new assembly system design can be realised; this dataset is represented by arrow 6.

- **Dataset 7** - Data from the assembly system or shop floor to the DES software and from workstations and components to the kinematic modelling software. This is essentially the feedback data from the implemented production line to improve the virtual model. This dataset is typically used in operational phase and the connection to DES is represented by arrow 6. The connection to kinematic modelling software is represented by arrows 4 and 5.

- **Dataset 8** - Data from kinematic model to DES software during the operational scale-up phase to create the new assembly system design. This data is used in the operational phase and is represented by arrow 7.

- **Dataset 9** - Data from DES to kinematic model regarding the new assembly system models that were created in DES. For example, the feedback regarding lack of productivity can be used within the kinematic modelling software to identify workstation designs that exhibit more productivity. This data is used in the operational phase and is indicated by arrow 7.

- **Dataset 10** - The final dataset is the data from the DES software to the assembly line or shop floor to commission the new operational phase scale-up design that was identified. This data is used in the operational phase and is indicated by arrow 6.

In the above datasets, it is assumed that when data from DES is communicated to the real assembly system, the associated workstation and component level data are communicated as well. The flow of data from concept phase to the operational phase for scale-up is discussed along with a detailed explanation of the datasets. Some of the data in the datasets might overlap with each other which implies that they are critical across various phases. Sometimes, the data that were communicated between the entities during the implementation phase might be the same type of data that is communicated in the operational phase. Therefore, the datasets are not mutually exclusive. The various data obtained from shop floor can be categorised into simulation relevant, simulation irrelevant, directly usable and indirectly usable [Mieth et al., 2019]. The data that are simulation relevant are considered within the context of this thesis. Based on this, the author has represented the datasets in **Figure 3.3** and **Figure 3.5** to represent the data.

Component data

Figure 3.3 shows the component and station data required for kinematic modelling software along with a brief description of each. Starting with the native component CAD, it is generally not in a format that can be directly imported into the kinematic model. It needs to be processed to lightweight CAD by removing unnecessary geometry information and converted to a suitable format depending on the modelling software used. Typically used formats include VRML and JT, however, most kinematic modelling software claim to support a number of formats. The processing to lightweight format involves the identification of dynamic and static elements of CAD; the dynamic elements are saved separately such that the kinematic behaviour can be associated with the geometry within the kinematic modelling software. The effort involved in importing the CAD is high due to the CAD processing. However,

		Importance		Data format			Data source				
		Core	Optional	Numerical	Graphic	String	CAD	Datasheet	Process planning software	Component library	Historical/empirical
Component data	Component CAD	✓			✓		✓				
	Weight		✓	✓				✓			
	Payload		✓	✓				✓			
	Torque		✓	✓				✓			
	Direction of motion	✓				✓		✓			
	Range of motion	✓		✓				✓			
	Accuracy		✓	✓				✓			
	Repeatability		✓	✓				✓			
	Energy consumption		✓	✓				✓			✓
	Cost		✓	✓				✓			✓
	Acceleration/deceleration profile		✓	✓				✓			
Station data	Station footprint	✓		✓			✓				
	Process sequence	✓				✓			✓		✓
	Safety interlock	✓				✓					✓
	Machine setup time		✓	✓							✓
	Energy consumption		✓	✓				✓		✓	✓
	Work instruction	✓				✓					✓
	Cost		✓	✓				✓		✓	✓
	Motion time of component	✓		✓						✓	
	State transition position	✓				✓					✓
	Layout	✓			✓		✓				

Figure 3.3: Component and station data for kinematic model.

if the components are built once, then they can be re-used in future models. Therefore, the effort involved reduces if pre-defined library models are used. The overall workflow in using CAD within kinematic model can be summarised as follows: geometry acquisition, dynamic parts identification and CAD processing, format conversion and model import, model assembly and kinematic behaviour definition and logical behaviour definition. This is followed by analysis within the software. Depending on which software is considered, these steps may vary slightly. Some software allow better user interface with options to drag and drop pre-defined library elements without the need to build the library from scratch. This is very useful, especially in concept stages. The component CAD data is simulation relevant data, but it is not available in a format readily usable within kinematic modelling software.

Within the kinematic modelling software, components mean the entities such as workstation frame, AGVs, human resource, etc. Workpiece-related information is also considered as component data. For example, the weight of the workpiece is very useful data that aids the selection of suitable grippers or actuators that can perform the assembly. Similarly, the weight of the gripper also aids the understanding of whether a robot can accommodate that particular gripper. On a similar note, the torque of some tools like nutrunners need to be known to verify whether they would be suitable for the considered set of process parameters. Similarly, the direction of motion of certain material handling units like conveyors and gantry units, helps investigate their suitability for the considered process parameters. Data such as the range of motion provides information about the maximum stroke distance of grippers and actuators to understand their suitability to assemble the considered product or workpiece. Sometimes, from sustainable engineering perspective, the energy consumption of workstations needs to be addressed, but it should be noted that the energy consumption data might not always be available and in such situations the effort involved is high because it needs to be calculated either manually or empirically. In summary, not all component level data might be necessary for building the virtual models since some of them are application-specific.

Station data

The required station level data for kinematic model are represented in **Figure 3.3**. The first data represented is the station footprint; it is used to select those equipment that fit an available workstation space. It is to be noted that different stations might occupy varying amounts of space in the shopfloor. The process sequence data represents the sequence of operations that need to be done to assemble a specific product in that station. The process sequence might change according to the product assembled, but the data is useful for station process time calculation and selection of suitable equipment to perform the process. Safety interlock data is essential for control code generation and understanding the dependency between different components of a station. An example where safety interlock data is used is the gripper modelling; when it picks up a part, the part should not be dropped while the gripper is in motion. For this purpose, safety interlock condition is created to prevent the gripper dropping the part abruptly. The machine setup time data is useful when

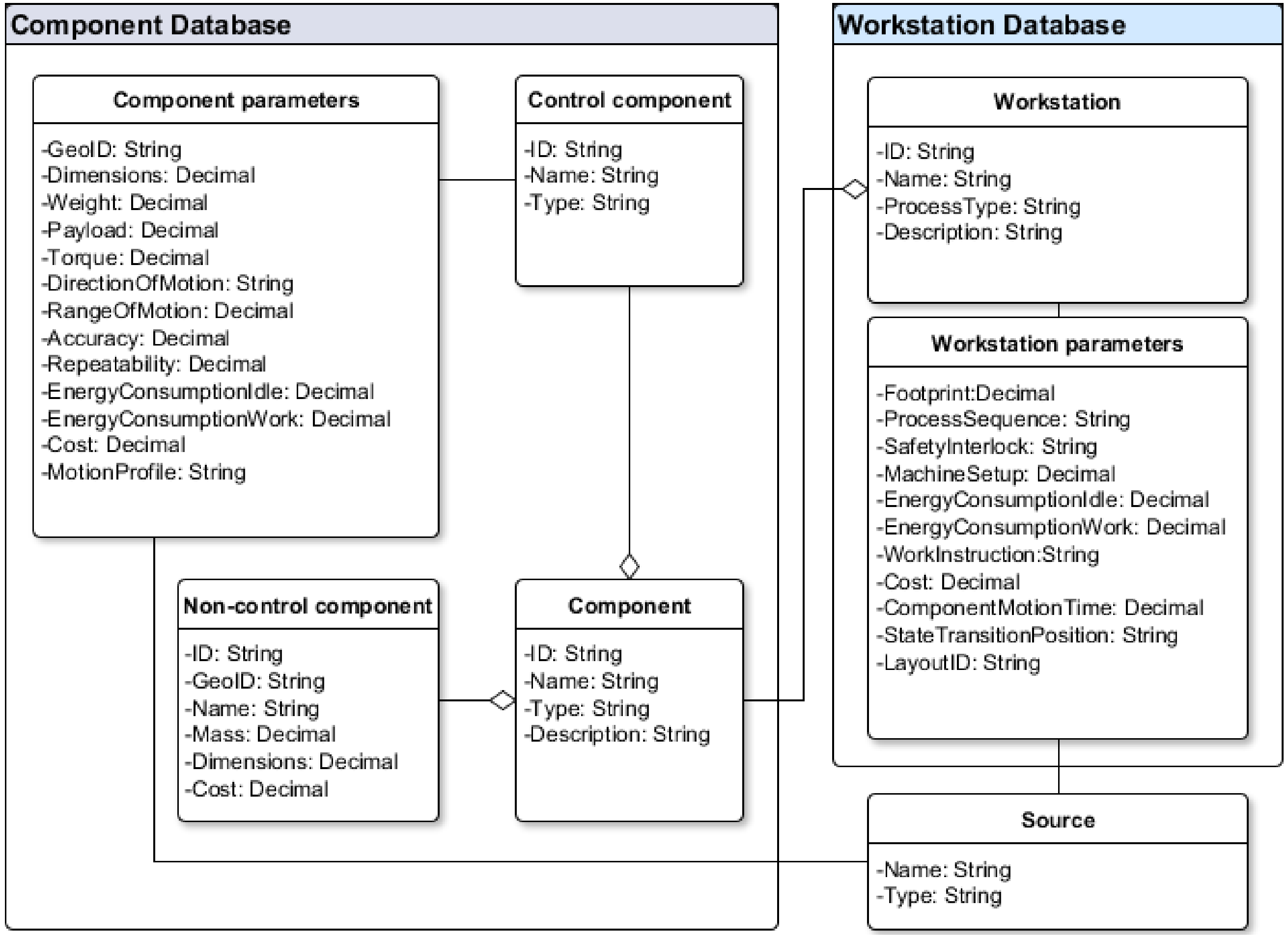


Figure 3.4: Component and station database schema.

different product variants are assembled in the same station. This setup time data further enables the process time evaluation and throughput analysis. The energy consumption data for the station can be calculated by using the energy consumption values of the individual equipment.

The energy consumption data can be used to filter equipment that meet the required standards of sustainability. The work instruction data refers to the sequence of tasks that are defined for the operator and this data allows ergonomic analysis and operator task time analysis. The motion time data of component is a crucial piece of information that can be obtained from historical data or empirical sources. If similar actuators, to what is intended to be used, were used in the past, the motion time data can be speculated from existing data. The motion time data can also be extracted from datasheets and it is essential to calculate the workstation processing times. Control components like actuator and gripper fingers might go through various states to advance from one location to another location and this is represented in the form of state transition diagrams. The state transition position refers to the location of the actuator or gripper relative to the coordinate system used in the software. This data enables process modelling, collision detection and path planning.

The layout data defines the way that stations and machines are arranged in the available floorspace. If layout data is already available from another software, it can be imported, otherwise, it has to be created again and this can be time consuming. To overcome this, it is possible to use point cloud data to automatically capture information about an existing layout. However, this is not feasible in the concept stage and hence there is need to rely on CAD layouts of the future assembly lines. From the sets of data that were investigated for the component and the workstations, a component and workstation schema is provided in **Figure 3.4** that explains the data along with their formats and respective categories. It is perceivable that the component is categorised into control and non-control component. This will be explained in more detail in section 3.3.1.

Input data for DES: control-related

The assembly system level input data required for the DES model is tabulated in **Figure 3.5**; they are categorised into logic, stochastic and control data. The control-

			Importance		Data format			Data source			
			Core	Optional	Numeric	Graphic	String	Historical/empirical	Datasheet	Kinematic model	Physical system
Control	Quantity	Operators	✓		✓			✓		✓	✓
		Workstations	✓		✓			✓		✓	✓
		Buffer capacity	✓		✓			✓		✓	✓
		Transporters	✓		✓			✓		✓	✓
		Batch size	✓		✓			✓			✓
	Layout	Floorspace	✓			✓		✓		✓	✓
		AGV control point	✓			✓		✓		✓	✓
		Transportation path		✓		✓		✓		✓	✓
		Operator guidepath		✓		✓		✓		✓	✓
		Equipment layout	✓			✓		✓		✓	✓
	Resource	Resource list and type	✓				✓		✓	✓	✓
		Resource capacity	✓		✓				✓	✓	✓
		Shift time and schedule		✓			✓				✓
		Sensor/RFID location		✓		✓				✓	✓
		Cost		✓	✓				✓		✓
		Energy consumption		✓	✓				✓		✓
		Resource allocation	✓				✓				✓
		Maintenance schedule		✓			✓				✓
Stochastic	Time	Process time	✓		✓			✓		✓	✓
		Setup time		✓	✓			✓		✓	✓
		Transport time/speed	✓		✓			✓	✓	✓	✓
		Operator travel time	✓		✓			✓		✓	✓
	Quality	Defect rate		✓	✓			✓			✓
		First time failure		✓	✓			✓			✓
		MTBF,MTTR		✓	✓			✓			✓
Logic	Scheduling	Station input output rule	✓				✓	✓		✓	✓
		Transporter input output rule	✓				✓	✓		✓	✓
		Resource dispatching rule	✓				✓	✓		✓	✓
	Routing	Product routing	✓				✓	✓			✓
		Transporter routing		✓			✓	✓		✓	✓
		Operator routing		✓			✓	✓		✓	✓

Figure 3.5: Assembly system data for DES model.

related data are further classified into quantity, layout and resource. The quantity data represents the number of operators, workstation, buffers, transporters and products that need to be considered in the DES model. The term *'transporters'* is used to indicate the various material handling units such as AGV, Automated Storage and Retrieval Systems (ASRS), conveyor systems, etc. The quantity of the resources is essential data to construct the DES model, to represent the system structure and to calculate the resource utilisation and cost functions. The data pertaining to the buffer capacity indicates the total number of parts that a buffer can hold and is used for capacity optimisation and operational performance evaluation. The last of the quantity data is the batch size which refers to the total number of products that are assembled as a batch; for mixed model production the batch size is one. This data is relevant for constructing the product flow and scheduling within the DES model.

The layout-related data consists of floorspace, AGV control point, transportation path, operator guide path and equipment layout. The floorspace data is essential for constructing the DES model, locating the various workstations within the available space and perceiving the total available space. It is also essential for reconfiguration, modifying existing DES models and layout optimisation. The AGV control point and path data needs to be indicated in the layout; this data can also be obtained from AGV fleet manager software. It is primarily used for layout optimisation and operational performance analysis. Operator guide path refers to the path or route taken by operators when transporting products, workpieces or raw material between stations; it needs to be defined within the layout. The equipment layout is an indication of where in the defined production system layout, the equipment will be commissioned or set up.

The resource-related data comprises of the resource list and type, capacity, shift-related information, sensor and Radio-Frequency Identification (RFID) location, cost and energy consumption data, resource allocation and machine maintenance data. Resource type refers to DES-specific details regarding the equipment used in the workstations and this can vary depending on the software used. For example, assembly machines that need to wait for certain set of raw materials before it can begin processing, or disassembly stations that separate the product into parts, etc., have specific behaviour that needs to be defined within DES to enrich the model.

Resource type also includes the various types of transporters depending on their behaviours and the various categories of operators according to their skill-level or certifications. The resource capacity refers to the number of products that a particular workstation can assemble simultaneously. It also refers to the total number of products that transporters and operators can handle simultaneously. Shift-related data refers to the shift pattern for all considered resources and it is essential for human resource utilisation analysis and cost analysis. When workstations or transporters have sensors or RFID tags attached to them, certain decisions are made with that data. For example, a pallet on a conveyor might have an RFID tag attached to it to read and check the status of the product on the pallet. Depending on the product status, the product route needs to be decided. This forms a core part within DES modelling logic and hence is simulation-relevant data.

Input data for DES: stochasticity-related

The considered stochasticity-related data are classified into time-related and quality-related. They are referred to as stochastic since the time and quality-related parameters, in reality, do not have a single value but rather take a value within a defined probability distribution. For the time-related data, the process, setup, transport time, operator travel time are considered. The process time considered here, is the time to produce or assemble a product in a workstation. However, within the context of DES, it does not include the setup time, which is the time taken to prepare or setup a workstation and it is more prominently used when more than one product variant is assembled. The reason that process time and setup time are separated in such a way is because they are considered as separate parameters in DES; but this might be software-specific. The last of the time-related parameters considered are the transport time and operator travel time; it refers to the time taken by transporters and operators to travel and transport the raw materials or workpieces from one location to another.

Considering the quality-related data, the defect rate, first time failure, Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR) are considered. The defect rate refers to the number of products that fail due to various reasons such as assembly or material issues. First time failure is the time that the machine that is in operation is expected to fail for the first time from the start of simulation run. MTBF represents the mean of the time between failures for the considered equipment. MTTR represents the average of the time to restore a failed equipment. All these data are necessary to perform maintenance-related modelling in DES.

Input data for DES: logic-related

The data related to the model logic are further categorised into scheduling-related and routing-related. The data regarding the input and output rules for station or transporter determines the source or location from where the product to be assembled is selected, in case of input rule or the location to where the assembled product is sent to, in case of output rule. The data pertaining to resource dispatching rule indicates how a specific operator or transporter is selected from a pool of available resources to do a particular job. For example, if there are ten operators available to perform an assembly at a specific station but only one is required, the rationale behind choosing one operator among the ten available is defined by these set of rules.

Among the routing-related data, the product routing determines the sequence of processes or stations that a product needs to go through in order to be assembled. In a similar fashion, the path taken by operators and transporters is defined in the operator and transporter routing respectively. Although the logic-related data are important for simulation, they form the most difficult set of data that hinder the automation of DES model creation; the effort involved in obtaining the data is also very high.

Output data from kinematic modelling software

The output data from the kinematic modelling software is provided in **Figure 3.6**. Some of these data, support modelling within DES, whereas some of them may not really be required for DES modelling. It can be seen that some of the data such as processing time, resource type, operator routing, transporter routing, number of operators, etc. which are mentioned in the **Figure 3.6** are also seen in **Figure 3.5** where the input data for DES are illustrated. This signifies the importance of the data obtained from kinematic modelling software. The data that might not be required within the DES software include PLC control code, detailed robot programming, collision detection, etc. The output data from kinematic modelling software helps system designers, control engineers and personnel involved in process planning and also helps work instruction generation. The simulation of the assembly process can be visualised and exported as video files that help explain the assembly concept. They play an important role in decision making at the workstation level, in the absence of a physical system. Additionally, they play an important role in control code generation and virtual (hardware-in-the-loop) commissioning and constructive commissioning [Lee and Park, 2014].

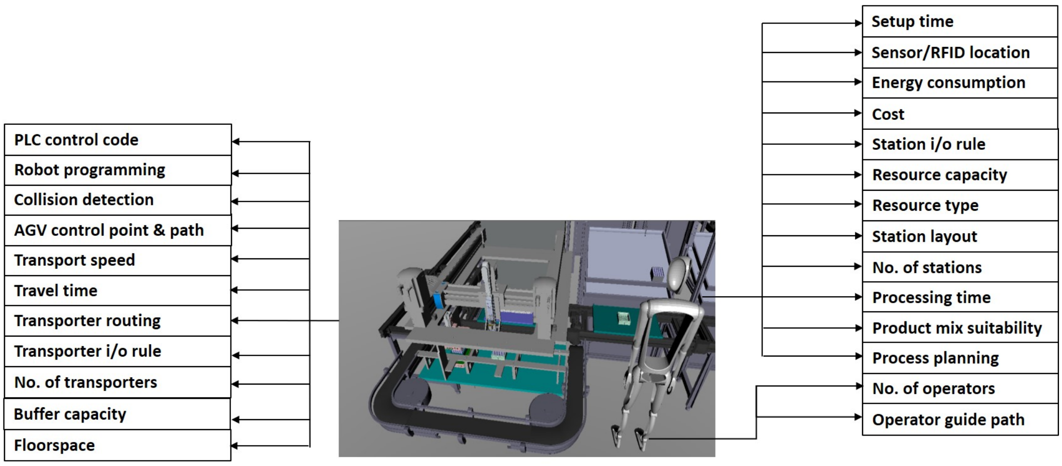


Figure 3.6: Output data from kinematic modelling software.

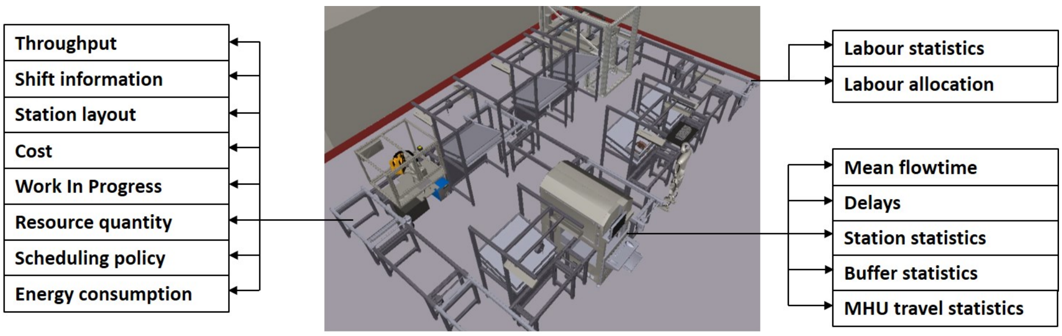


Figure 3.7: Output data from DES software.

Output data from DES

The output data from DES are presented in **Figure 3.7**. It should be noted that the presented set of data is only a representative list and not an exhaustive one. Majority of the data from the DES software are the results of *'what-if'* analysis and they provide decision support for different scenarios that could potentially occur. Data such as throughput, mean flow time, resource utilisation and work-in-progress are useful for scale-up analysis. This is because they indicate the productivity of the different scenarios considered for scale-up. Without these indicators, it becomes difficult to choose a specific solution amongst the many available possibilities. From simulation-optimisation perspective, the output data from DES plays a critical role in calculating the objective function; especially data related to cost and productivity are considered important; a number of manufacturing system optimisation problems consider them as part of their objective functions [Prajapat and Tiwari, 2017]. This set of output data from DES software is useful for decision making and can be passed from the DES model to the system designers and decision makers using dashboards, reports or other visualisation software.

The investigation of data that are required by simulation tools, especially kinematic modelling and DES, at various stages of the lifecycle and the use of the data and its importance for simulation was presented. From the analysis, some critical data that are used across multiple phases and software are the processing time, layout, operator and transporter path, and resource quantity. Since the data are shared across various software, the next section explains the methods for exchange of data.

3.2.5 Data exchange

This section provides a brief write-up of the data transfer between different software that will be discussed in the methodology. Two options for data transfer were considered as shown in **Figure 3.8**. In option 1, the software communicate with each other either directly or with the help of an interface without the use of a common data repository. On the other hand, in option 2, the software communicate with each other through a common data repository; this repository could be a database. A database, typically comprises of related data which are deposited in one location. For the DDSM methodology, option 2 will be adopted and a common database will be used for storing and retrieving the data. The reasons for selecting option 2 can be summarised as follows: i) storage of necessary data in a centralised hub, thereby enabling integration of data from heterogeneous sources [Boucelma et al., 2002]

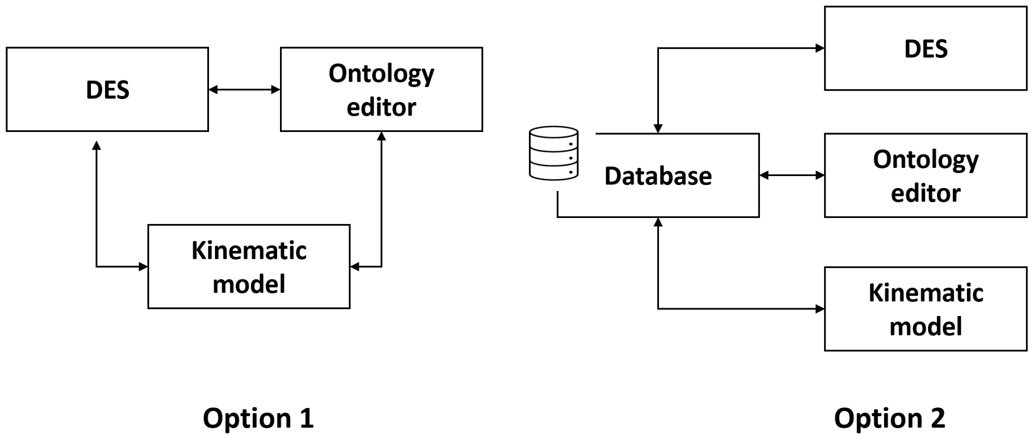


Figure 3.8: Data exchange options.

ii) to avoid descriptive conflicts [Tolk and Muguira, 2003] and duplication of data [Al-Najjar, 1996], and iii) achieving a holistic view of the simulation-relevant data for scale-up [Kans and Ingwald, 2008]. In the DDSM methodology, all data will be provided with a unique *'identifier'* to avoid duplication. There are many ways to establish connection with the database. It can be done manually or automatically, by direct data transfer or indirect data transfer using Application Programming Interface (API). The following sections will discuss the data exchange standards and formats.

Various standards and formats for data exchange have been proposed in literature. One such standard is the **Core Manufacturing Simulation Data (CMSD)** and this standard will be adopted for the DDSM methodology. CMSD is chosen for this research because of its suitability to provide interoperability between manufacturing system and simulation data. One of the core benefits of using CMSD is to allow for the automation of data transfer which eliminates human errors during data input [Bloomfield et al., 2012]. CMSD provides a neutral data format for the interoperability between simulation software and manufacturing system and the specification is available in two different methods: **Extensible Markup Language (XML)** and **Unified Modelling Language (UML)**. Simulation Interoperability Standards Organisation (SISO) proposed the standards for CMSD UML format in 2010 [CMSD, 2010] and XML representation in later in 2012 [CMSD, 2012] released the XML

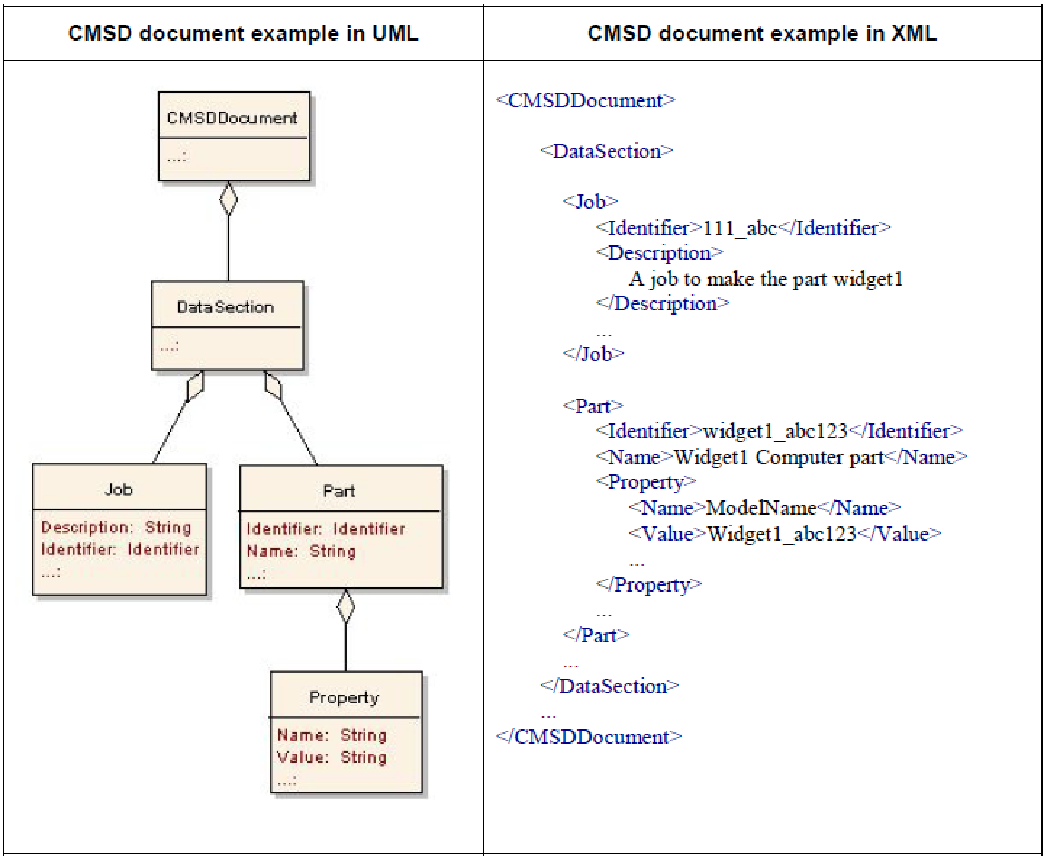


Figure 3.9: Example of CMSD document adopted from SISO [CMSD, 2012]

representation. An example of the CMSD document is depicted in **Figure 3.9**.

From the standard [CMSD, 2012], the CMSD model does not focus on all aspects of manufacturing; the main areas of focus are process planning, scheduling, inventory management, production management and plant layout; it comprises of six packages: i) layout, ii) part info, iii) support, iv) resource info, v) production operation, and vi) production planning. The layout package consists of classes and relationships that are associated with spatial representations and manufacturing layout. The part information package consists of classes associated with part-related data such as batch size, work in progress raw material, finished products, etc. Production operation class comprises of order status, scheduling elements, processing time, parts produced, etc. The resource information is associated with equipment characteristics, operator skills, setup information, etc. The production operation is associated with shift schedule, process plan, maintenance plan, etc. For the DDSM methodology, however, not all of these packages might be utilised. The data exchange format is explained in more detail in the latter part of section 3.3.1.

The data exchange and flow between the various software modules in DDSM is highlighted in **Figure 3.10**. This schematic is obtained from the investigations and analyses performed in section 3.2.4. Each alphabet in **Figure 3.10**, is assigned to a set of data that is transferred or exchanged. **Set (a)** represents the data that is passed to the common database from kinematic model to be used in the knowledge representation module or ontology editor. This set of data enables the query design within the ontology editor and comprises of the following: i) workpiece attributes such as dimensions, weight, product identifier, shape, gripping surface area, ii) process related information such as sequence, number of processes, task types in each operation, degree of freedom, axes of motion required and other process-specific parameters, and iii) resource-related data such as station footprint and allowable weight. The **set (b)** comprises of data from the knowledge representation module or ontology editor that is passed back to the common database. This is used within the kinematic modelling software for validating the query results; the results represent a group of equipment that have the capability to perform the required processes. Example of data in set (b) include equipment data such as the attributes, tasks performed, motion time, cost and energy consumption, etc.

The workstation configuration data are represented in **set (c)** and they are passed from kinematic model after the validation to the common database. Typically, the workstation KPIs such as the workstation process time, energy consumption, cost, etc., are part of set (c). This data is necessary to improve the quality and accuracy of the assembly line models in DES. **Dataset (d)** represents the workstation data transferred from the common database to the optimisation module for calculating the objective function. As an example, consider a cost function defined in the optimisation objective which comprises of various elements of cost. These elements of cost for the considered workstation configurations are obtained from the common database and forms part of set (d). Depending on the formulation of the objective function, the data that will be required varies.

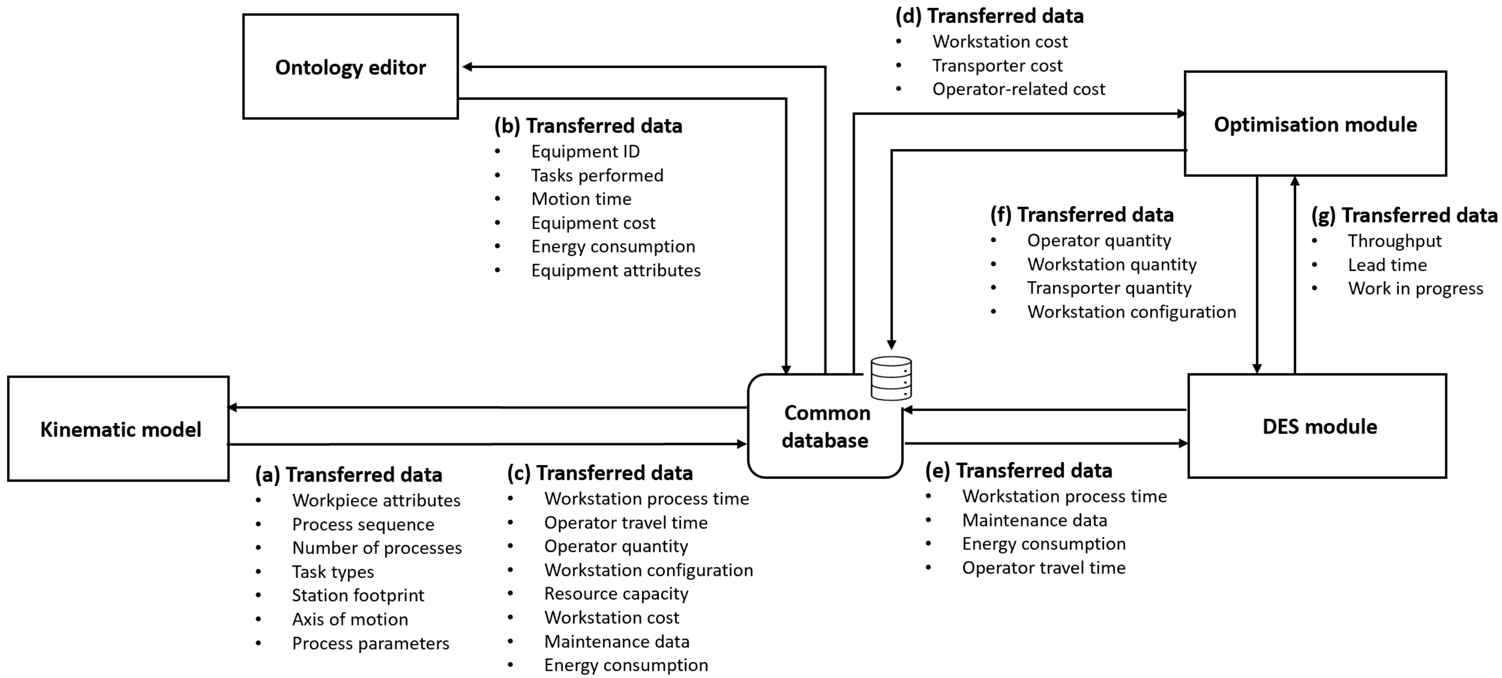


Figure 3.10: Schematic of data exchange and data flow in DDSM.

The **dataset (e)** consists of data such as the workstation process time, maintenance data, operator travel time data, etc. This set is passed to DES software from the common database and is necessary to perform good calculations and decisions within DES. This set of data originally arises from the workstation configurations that were validated within the kinematic model and later stored in the common database. **Dataset (f)** refers to the decision variables that are passed from the optimisation module to the DES model. The decision variables are those that need to be optimised and their values are obtained from the optimisation module. They play a vital role in simulation optimisation. **Dataset (g)** represents data such as the throughput, lead time, work in progress, etc., that are passed from the simulation back to the optimisation module. This dataset plays a vital role in calculating the objective function that will be detailed in section 3.5. This summarises the data that are exchanged or transferred between the different software and their role in the DDSM framework. The exact data format that will be used and the way the connection will be established is discussed in the following section.

3.3 Stage one: Workstation Configuration Selector

Stage one of the methodology comprises of two modules: kinematic modelling module and knowledge representation module. **Figure 3.11** provides a detailed view of Stage one of DDSM methodology. The kinematic modelling module corresponds to the kinematic modelling software and its primary objective is the analysis of the process sequence, parameters, constraints, etc., of the existing virtual model of the production line prior to the modifications. The kinematic modelling module is coupled with the knowledge representation module which is built upon a Product Process Resource Resource attribute (PPRR) framework. The knowledge representation module comprises of an ontology editor; using the knowledge mapping between PPRR, the candidate equipment list will be queried to identify those equipment that can perform the required operations. Both modules communicate through the common database and the results of the analysis provides the the workstation designs, the constituent equipment and the workstation KPIs that can be used for comparing the alternative designs.

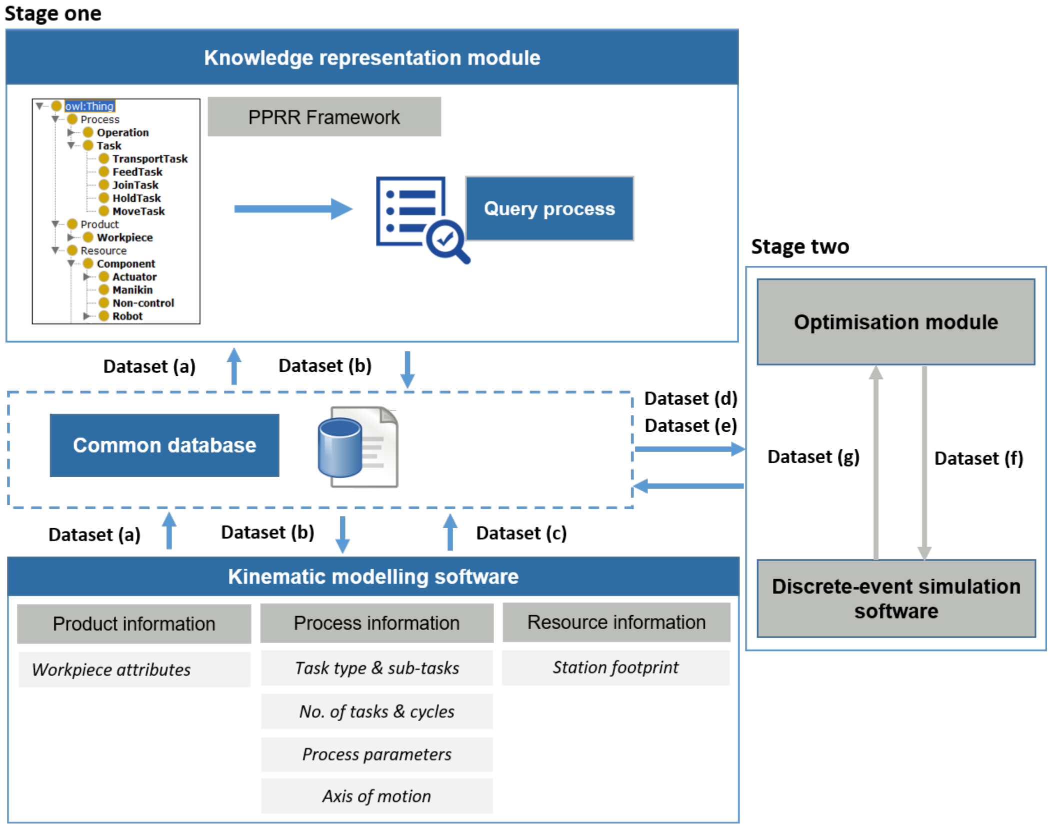


Figure 3.11: Stage one of DDSM methodology.

3.3.1 Kinematic modelling module

The kinematic modelling module mainly comprises of the kinematic modelling software which is typically used to model and visualise production systems, primarily for path planning, clash detection and verification of assembly process in the absence of a physical system [Maropoulos and Ceglarek, 2010]. Due to their ability to model the kinematics, they can predict the workstation processing time [Caggiano and Teti, 2018]. The workstation processing time data from the kinematic model can be leveraged to increase the accuracy of DES models [Chinnathai et al., 2019] . The input data from the shopfloor for use within this module were highlighted in **Figure 3.3**. Within the kinematic model, the modelling can be done at two abstraction levels: the component level and workstation level. The component model encapsulates information regarding the various components such as robots, gantries, grippers, etc. The workstation model encapsulates within it several component models. For example, a welding station model can have several components such as weld gun, welding robot, fixtures, etc. Moreover, process-related modelling is also done at the workstation level. The following three steps form part of the initial workflow in this part of the methodology.

1. Create component model of the pilot or production line components within the kinematic modelling software.
2. Combine the component models to create the workstation model within the kinematic modelling software.
3. Investigate the relevant process-related data that will form part of **dataset (a)** which will eventually be passed to the knowledge representation module through the common database.

During this data exchange between the kinematic modelling software and knowledge representation module, data such as the process parameters, constraints, process sequence, machine setup, etc., are passed from the kinematic modelling module to the knowledge representation module. At this stage, the various product, process and resource elements such as workpiece, processes, AGVs, conveyors, etc., are referenced with an *'identification tag'* that is unique to them. After necessary analyses are performed in the ontology editor, the selected equipment and their attributes are passed back to the common database. Following this, they are validated and their performance is tested. A kinematic modelling tool developed by the Automation Systems Group in the University of Warwick, known as vueOne, is used for the purpose of kinematic modelling in DDSM; more detailed explanation of vueOne is provided in the next section.

VueOne engineering tool description

For the purpose of kinematic modelling, **VueOne** Engineering toolset is used in this thesis. The VueOne toolset was built by the Automation Systems Group (ASG), University of Warwick. VueOne was originally structured upon component-based modelling approach [Lee et al., 2007, 2005] and it fits well within our bottom-up two stage methodology. Due to this reason, the DDSM framework can be considered as a component-based bottom-up modelling approach. The vueOne software enables the use of component library. This allows faster building of systems using the pre-built components or modules. Additional benefits include the shortening of planning stage and encapsulation of the physical system as a virtual model. [Müller and Horbach, 2012] consider two types of components: object and planning. The object components represent the structural entities of a factory and the planning component represents elements associated with the planning process.

The toolset supports the lifecycle using the following applications: process planning, system reconfiguration, control code generation, basic ergonomic analysis and virtual commissioning. Additionally, several extensions to the capability of the software have been proposed over the years [Ghani, 2013; Alkan, 2018; Ahmad, 2017; Mus'ab H, 2017]. VueOne is intended to be a lightweight virtual engineering tool that uses the standard XML format for data exchange. The lightweight models are characterised by the reduced level of detail in the number of polygons, simplified geometry features and reduced number of parts. This is achieved by pre-processing the component CAD before its use in vueOne and converting the format from the native CAD to a VRML format. During this transformation, unnecessary data such as nuts, bolts and screw which do not add any value to the required objective are removed and the geometry is simplified. The movable elements of the component CAD are identified and saved as a separate file. The kinematics of the movable elements are then defined within the kinematic model. Additionally, the transparency, color and other visual aspects of the CAD elements can be modified as required during the processing stage.

Figure 3.12 shows the vueOne component initialisation and library. The workflow in the kinematic modelling process starts with the component modelling where the geometry of the structural elements of the assembly line are selected and modified as required. The component library provides a list of all existing component models in the software. It is possible to add as many models as required.

VueOne: Component modelling

The modelling process within the software involves two main platforms: **component** and **system/workstation**. In component modelling, the first step is to define

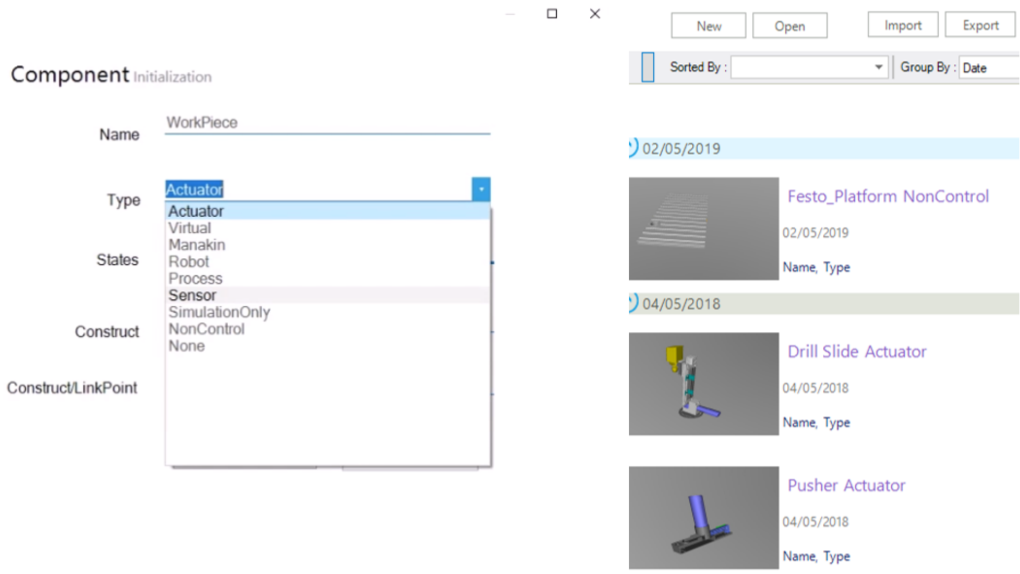


Figure 3.12: vueOne component library.

the *'type'* of the component that will be modelled. The *'type'* determines the behaviour of the component. For example, the component might be an actuator, robot, operator model, etc. Once this is determined, the next step is to visualise the geometry. The geometry of the components is imported within the software from the **3D model editor** module and an example of the geometry library is shown in **Figure 3.13**. Following this, the behaviour of the component needs to be modelled. In this step, the various component-related data such as the payload, mass, range of motion and energy consumption that are mentioned in **Figure 3.4** are added as *'parameters'*. This step is important for the encapsulation of component-related data and for re-use of the component models for other related applications.

VueOne: Non-control component

Two main categories of components are modelled in vueOne: control and non-control components. The term *'non-control component'* refers to the entities such as workstation frame, station platform, support elements, etc. They are passive components that are not directly involved in the assembly processes and are generally stationary. However, it is important to model them since they provide visualisation

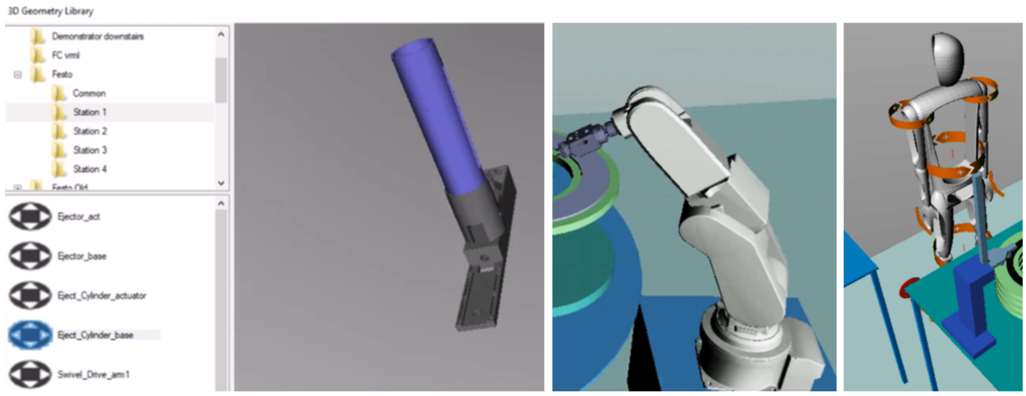


Figure 3.13: vueOne geometry definition, V-Rob and V-Man.

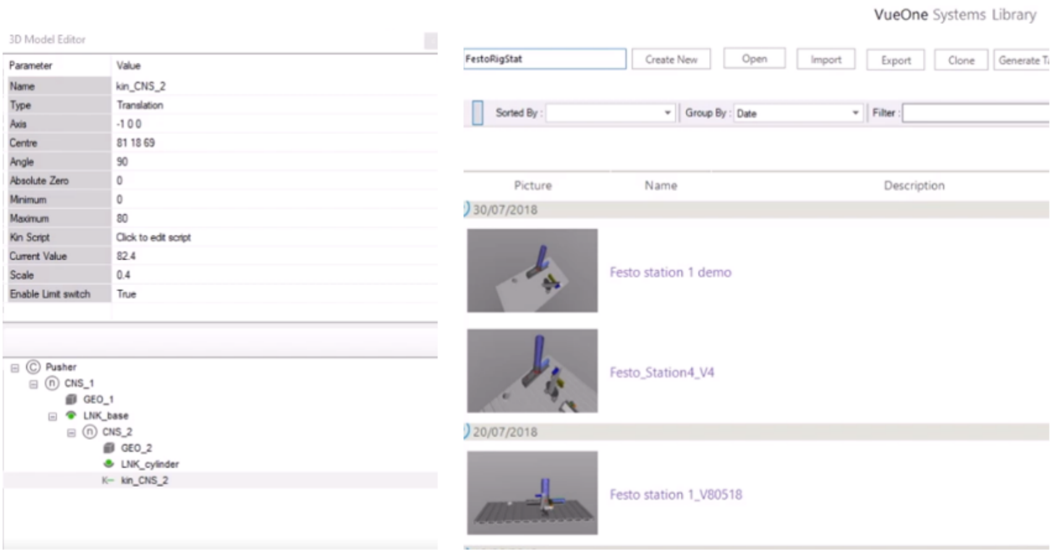


Figure 3.14: vueOne kinematics definition and system library.

and support the analysis of necessary workstation features such as station footprint and weight capacity which are useful to compare the workstation configurations.

VueOne: Control component

The term *'control component'* refers to entities such as grippers and robots. They are typically involved in the assembly process and exhibit kinematics behaviour. Control components can be further classified into actuators, robots, human model, sensor, etc. The movable elements of the component that were separated during

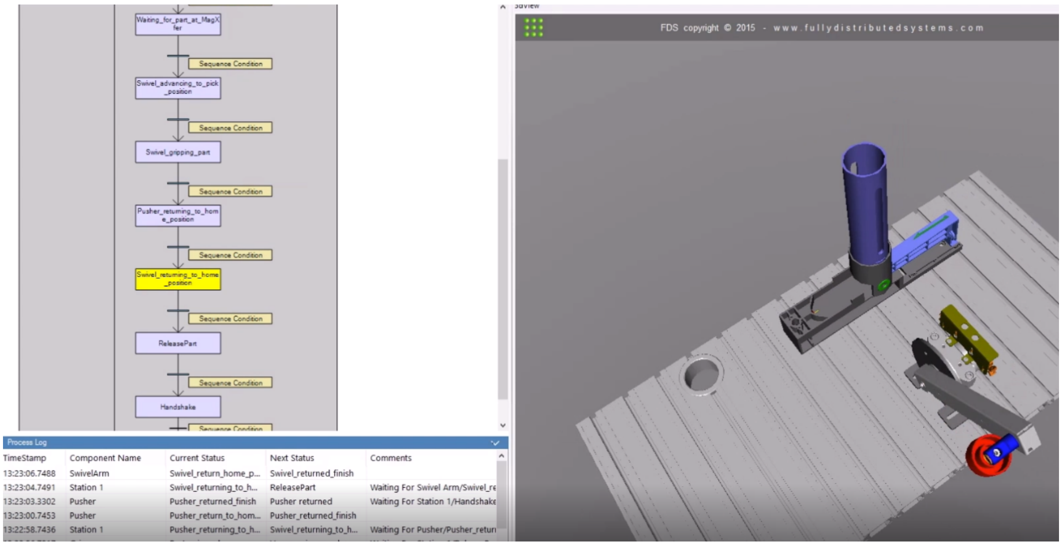


Figure 3.15: vueOne process flow and system model.

CAD processing are modelled as actuators as they are associated with control logic. The control logic is defined with IEC-61131-3 compliant State Transition Diagrams (STDs) [Ahmad, 2017]. These states can be static, dynamic or initial; initial represents the home position. The static and dynamic states represent the stationary and moving states, respectively. As an example of modelling control components, consider a *'pusher'* consisting of two elements: frame and an actuator to push the workpiece. **Figure 3.14** shows a screenshot of the 3D model editor for a pusher. As seen from the Figure, the two elements are defined as two *'constructs'*, CNS_1 and CNS_2 within vueOne; they are linked together using two *'linkpoints'*, LNK_base and LNK_cylinder at specific co-ordinates. After this, one of the two types of kinematic behaviour, translation or rotation kinematic needs to be defined. The *'pusher'* will have to push the workpiece from location A to B and for this purpose a translation kinematic, *'kin_CNS_2'*, is defined. The *'axis'* option in the *'3D model editor'* is used to define the direction of motion and the range is defined using the *'maximum'* and *'minimum'* options. The *'current value'* option provides the current distance travelled by the actuator. In this way, the actuators and other control components are virtually modelled.

VueOne: V-Rob and V-Man

In vueOne, human modelling and robot modelling have their unique modules named V-Man and V-Rob, respectively and an illustration of this is provided in **Figure 3.13**. The V-Man module utilises MODular Arrangement of Predetermined Time Standards (MODAPTS) for determining the operator task time values. By defining the process that is being done based on work instructions, the time that an operator will take to perform a process can hence be calculated. Additionally, the V-Man module allows the analysis of basic ergonomics which flags up specific instances and issues that need to be analysed further, possibly with another software. The V-Rob, in a similar fashion, allows the determination of the time taken to perform certain activities by robots. V-Rob has an inbuilt library of robots that can be used whenever the necessity arises. Additionally, there is the possibility of adding more robots as and when necessary. By combining V-Man and V-Rob activities, it is possible to model manual, automatic and semi-automatic workstations. In this manner, the various components such as two-finger grippers, three axis gantries, six axis robots, etc., are stored in the component library. Each component is assigned a unique ID that will remain with the component throughout the lifecycle.

VueOne: System modelling

The next step is the system modelling and an illustration of this is provided in the **Figure 3.15**; the first activity is to import the required components from the library. These components are the ones that were created during component modelling. They are then assembled together using linkpoints; this is followed by definition of the logical behaviour, process sequence and process flow. The process sequence determines which components need to perform the required processes and in what order they perform; this further dictates the relationship between the various components of a system. The relationship between the components are defined using the (State Transition Diagrams) STDs. It is also possible to use sensors to signify decision points in the model. The product that is assembled or manufactured is referred to as the workpiece and it can be picked up and placed at various locations within the model using workpiece linkpoints.

Once the system is constructed and the logical behaviours are described, the process

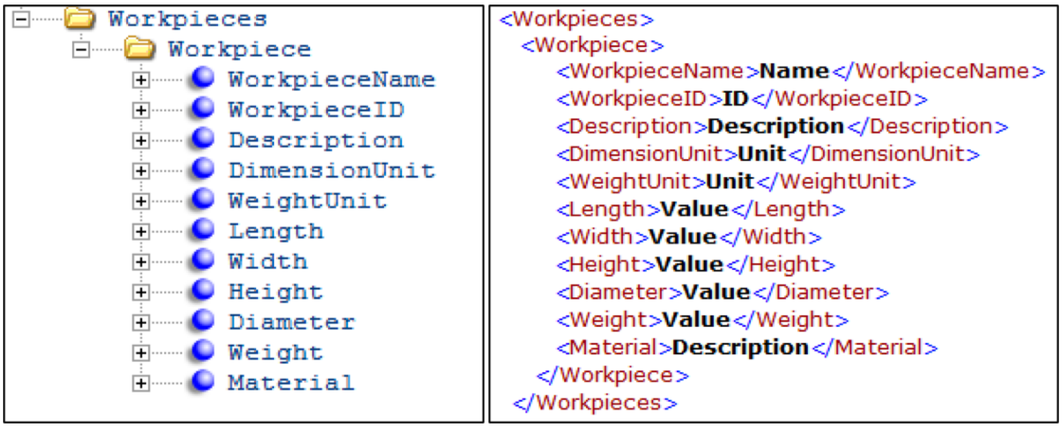


Figure 3.16: Workpiece data from kinematic model to database.

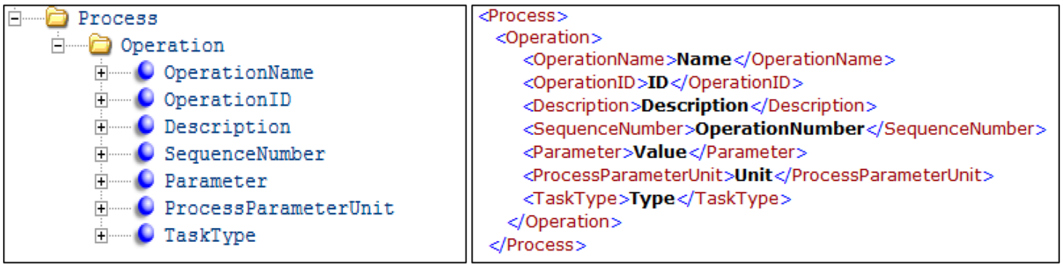


Figure 3.17: Process data from kinematic model to database.

time analysis, collision detection, path planning, control code generation, virtual commissioning, comparison of concepts, etc., can be done. Specifically, in DDSM, process time analysis and comparison of different workstation design concepts are the primary targeted activities. On completion, the model can be exported to XML format such that the data can be utilised in another software. Using the CMSD standard as the base, XML schemas for the workpiece information, resource information, process information that will be passed from kinematic modelling software to the common database is provided in **Figures 3.16, 3.17, 3.18**, respectively.

3.3.2 Knowledge Representation module

The knowledge representation module is designed using **Protégé**, a free open source ontology editor developed by Stanford Centre for Biomedical Informatics Research [Knublauch et al., 2004]. It was selected due to its wide and active user community, accessibility and availability of support [Jain and Singh, 2013; Knublauch

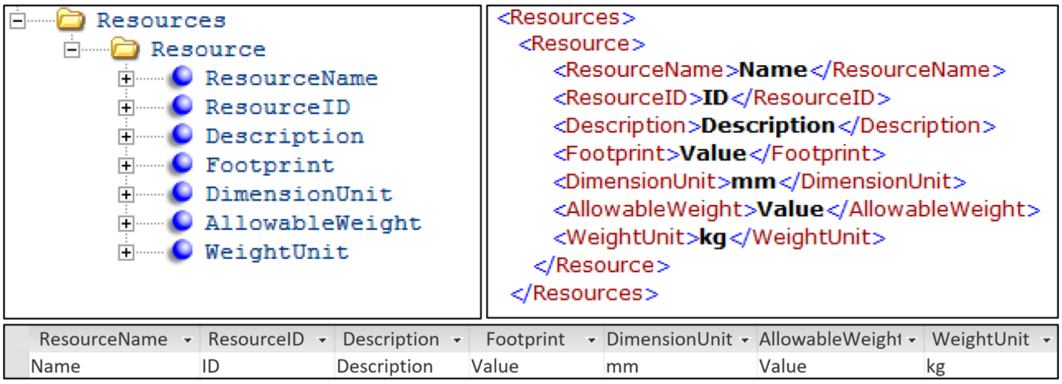


Figure 3.18: Resource data from kinematic model to database.

et al., 2004]. Ontology, as explained by Gruber, *"is an explicit specification of a conceptualization"* [Gruber et al., 1993]. The reasons for using ontology can be summarised as follows: i) providing people and software a shared understanding of concepts and terminologies, ii) for knowledge reuse and analysis, iii) to store collections of data and query its contents for information retrieval [Uschold and Gruninger, 1996], and iv) to achieve data mapping between heterogeneous software [Penciuc et al., 2014]. Having presented the benefits of using ontology, the following brief write-up explains the need to employ ontology for this particular research. The DDSM methodology, in Stage one, endeavors to generate potential workstation configurations by retrieving suitable candidates from an existing catalogue of equipment that meet the process requirements. Additionally, considering the fact that the manufacturing system is comprised of the physical existing entities, it is suitable to use ontology, which typically deals with the study of existence and relationships, for specifying and mapping the workpiece, equipment and their relations. Moreover, an ontology-based approach is considered suitable for representing complex manufacturing systems [Lohse et al., 2005].

In this paragraph, two relevant ontology models for manufacturing systems are discussed. The first one is a PPR ontology adapted from [Ferrer et al., 2016, 2015], wherein product attributes are mapped to process and resource concepts and integrated with a kinematic modelling software. The application area is to improve the product, process and resource modelling of assembly automation systems. The second work presents a holistic equipment ontology incorporating a Function Behaviour Structure paradigm for equipment selection [Lohse et al., 2005] for Recon-

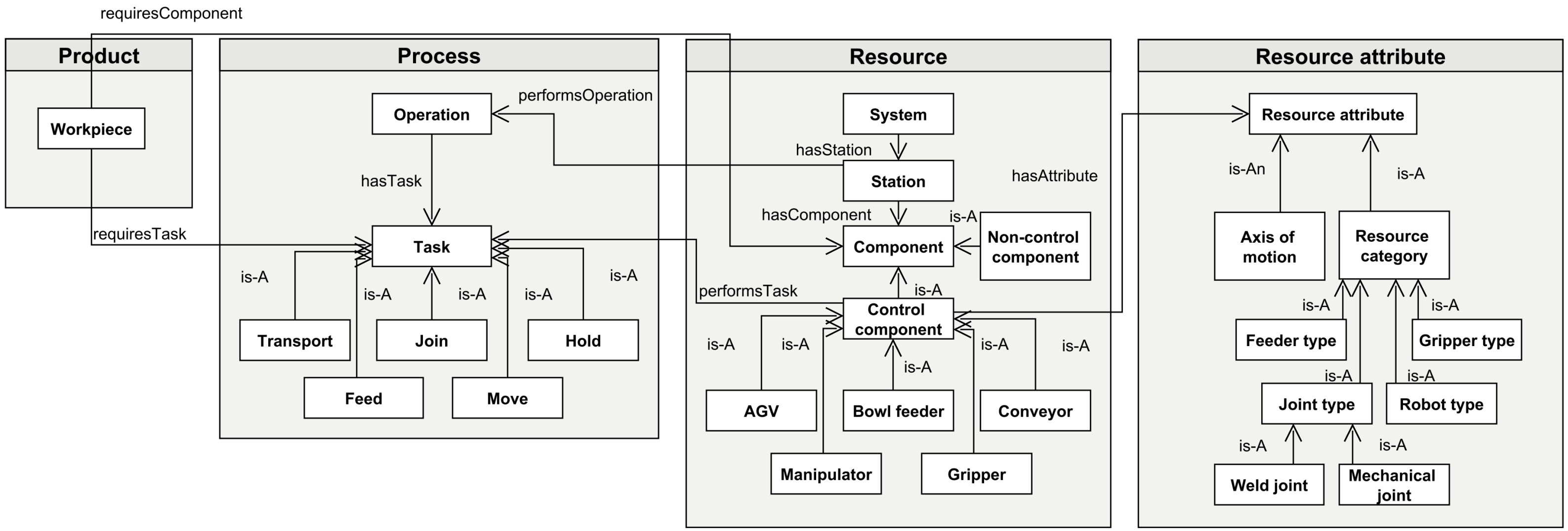


Figure 3.19: DDSM ontology structure in Protégé.

figurable Assembly System design and re-engineering process. The main objectives are to select suitable equipment, configure the selected equipment and evaluate alternate configurations to select the most suitable one. In their framework, the equipment is further decomposed into system, cell, workstation, unit, device and element. The equipment is linked to its functions and behaviours. Function determines what the assembly process does and behaviour and structure together determine the property and performance of the equipment. The equipment, through its behaviour, fulfils the required function.

Although both ontology models focus on assembly systems, the PPR framework considers the integration with kinematic model and is more relevant for the DDSM methodology. Therefore, the original PPR framework is adapted and modified to the **PPRR framework** to support the equipment selection process in DDSM. The structure of the ontology is presented in **Figure 3.19**. Additionally, the structure of the PPR ontology framework correlates with the structure of the vueOne kinematic modelling software; the vueOne component architecture comprises of control and non-control component and the PPR ontology model represents this. However, the ontology model used in the DDSM methodology differs from the mentioned articles in that i) it pursues the objective of supporting system configuration selection for transition from low-volume to high-volume, ii) it has a *'resource attribute'* class which represents the resource attributes that need to be queried for selection of suitable equipment, and iii) it provides a comprehensive list of resource data properties such as payload, weight, working range, dimensions, gripping force, accuracy, torque, maximum thrust, etc., that relate to process parameters for assembly operations. Another point to note is that the DDSM ontology architecture isn't necessarily bound to system reconfiguration but also considers the commissioning of new facilities and replacement of existing workstations if they are found unfit for purpose. A summary of the key features of the ontology model in DDSM are provided in **Table 3.1**.

Product and process class

The PPRR framework comprises of product, process, resource and resource attribute classes as explained in **Figure 3.16**. The product class comprises of a

Table 3.1: Features of DDSM ontology model.

Ontology feature	Description
Structure of ontology	Product, process, resource and resource attribute classes
Application area	Workstation configuration selection
Integrated software	Kinematic modelling software through database
Preferred query engine	Semantic Query-enhanced Web Rule Language
Reasoner	Pellet

'workpiece' or *'part'* that is mapped to a resource as well as the required assembly process. The process class comprises of two elements: operation and task. Each workstation is linked to one or more operations. The operations are composed of tasks that are the elementary actions that cannot be further sub-divided. They can be derived from the process sequence that can either be obtained from the kinematic module or the production system. Within the kinematic module, the process sequences are represented in the State Transition Diagrams.

Task subclass

For the purpose of this research, five task types adopted from [Chinnathai, Alkan and Harrison, 2017], move, hold/release, feed, transport and join are considered for the query process and they are represented as five instances that belong to the *'task'* subclass. When they are defined as instances, they become individuals or members of the *'task'* subclass. In Protégé, an instance can be mapped to another instance using *'object property'*. It is important that the equipment in the catalogue that are considered for the selection process are also defined as instances such that they can be mapped to the respective instance of the task subclass. For example, consider a hopper that feeds parts into the line. To model this in Protégé for DDSM, the hopper needs to be an instance of the resource class and the *'feed task'* needs to be an instance of the task subclass. Following this, the mapping between the hopper and feed task is established using the *'performsTask'* object property. It is possible for one equipment to be mapped to more than one task instance. The instance definition for resource class is explained in detail in the next paragraph.

Resource class

The resource class is sub-divided into system, station and component sub-classes in increasing order of granularity; a system is built up of stations and stations are built up of components. The term component here refers to the equipment such as weld gun and robots that are used to perform the various tasks. Components are further subdivided into control and non-control component, as explained in **Figure 3.4**, depending on whether they have logical behaviour or not. This is in alignment with the modelling architecture of vueOne.

Five types of control components are considered in the resource class: the gripper, Automated Guided Vehicle (AGV), manipulator, bowl feeder and conveyors; the components may or may not differ in the type of tasks that they perform. A specific component such as a *'two-finger gripper'* from brand *'XY'* can be added as an *'instance'* to the gripper subclass. In this way, the various components are added to their corresponding subclasses as *'instances'* and mapped to one or more of the defined five tasks using the *'performsTask'* object property. To illustrate this, consider a robot *'ABC'* capable of performing the *'move'* as well as *'feed'* tasks; robot *'ABC'* is an instance of the *'robot'* subclass and *'move'* and *'feed'* are instances of the *'task'* subclass and *'ABC'* is mapped to the two tasks using the *'performsTask'* object property.

Data properties are used to map an instance to a specific type of data that can be a real number, integer or string. The values of data properties such as range, dimensions and payload of the resource elements can be obtained from various sources. Consider a pneumatic gripper named *'GAXF1'* having a payload of 500g; it is an instance of the *'gripper'* subclass. To map the gripper to the value of 500g, the data property *'hasPayload'* is used.

Resource attribute class

The novel *'resource attribute'* class consists of two sub-classes: *i*) axis of motion and *ii*) resource category, which includes the robot type, joint type, gripper type and feeder type. This information is useful to enrich the workstation configuration selection process by screening the resources that possess the desired behaviour and category. The reason behind adding the attributes as a separate class is to have the attributes as instances and not as data properties. This enables mapping them to equipment instances using the object property. For example, consider a *'four axis robot'* that is added as an instance to the *'manipulator'* sub-class. The information regarding the axis of motion is important for the selection process and hence it is necessary to link the robot to this data. There are two options to do this: i) add the information about the axis as a data property or ii) add the information about the axis as an instance to the resource attribute class. The axis of motion can be classified into x,y,z, for translation and a,b,c for rotation. It is hence limited to six values which makes it possible to add it as an instance that can be mapped to the equipment instances. On the other hand, adding them as a data property means that the value needs to be entered every time an equipment is defined. Therefore, the option of adding them as an instance saves time since they do not have to be added to each and every equipment data property but could instead be selected from the existing six values.

Similarly, considering the resource category, the various resource types can also be added as instances instead of data properties since the categories are limited. It is to be noted that, certain equipment properties such as payload, length, height, etc., can take up a value from a number of possible values and it does not make sense to add them as instances. The resource category comprises of four instances: feeder type, gripper type, joint type and robot type. Each of them is explained in detail in the next paragraph.

The *'feeder type'* instance considers feeders that do bulk feeding and those that feed individual components or parts. It might be important to choose a feeder that feeds individual components in situations where the components are fragile or safety critical such as batteries. The *'joint type'* instance considers the equipment that perform welding or mechanical joining. This can include specific methods such as pulse arc welding, friction stir welding, brazing, etc. The *'robot type'* instance considers the various categories of robot such as SCARA, cartesian, six-axis, cylindrical, delta, co-bots and mobile manipulators. The *'gripper type'* instance considers the different mechanisms of gripping such as vacuum, hydraulic, magnetic, etc. The resource categories, in overall, is intended to allow for better querying and selection of equipment.

Protégé workflow

The workflow in the knowledge representation model is represented in **Figure 3.20**. The flow starts with the product and process data from kinematic model and is followed by the analysis of the process sequence and number of operations. The tasks that belong to each operation are verified to understand whether they belong to one of the five defined task types. If they do, then the equipment that can perform the tasks belonging to that operation are identified within Protégé as explained in **Figure 3.20**. Within Protégé, the process requirements for each operation are first considered and translated to parameters that are used to screen the existing set of components using *'query language'* to identify suitable equipment. Once the equipment are identified, the equipment along with their attributes are passed to the kinematic model through the common database. They are later modelled in the kinematic modelling software for validation. During the validation process, the equipment are combined to form the workstation configurations and these configurations are represented as a design table in the database which lists the various workstation configurations along with their KPIs; an example of the workstation design table is provided in **Figure 4.14**. Considering decision point in **Figure 3.20**, if an operation has tasks that do not belong to the defined five tasks, the operation will be ignored and the next operation in sequence will be subjected to the same procedure. As seen from **Figure 3.20**, the equipment selection is done for all the operations in the sequence after which the workstation configurations can be generated with the selected equipment. A detailed example of this is provided in section 4.2.2.

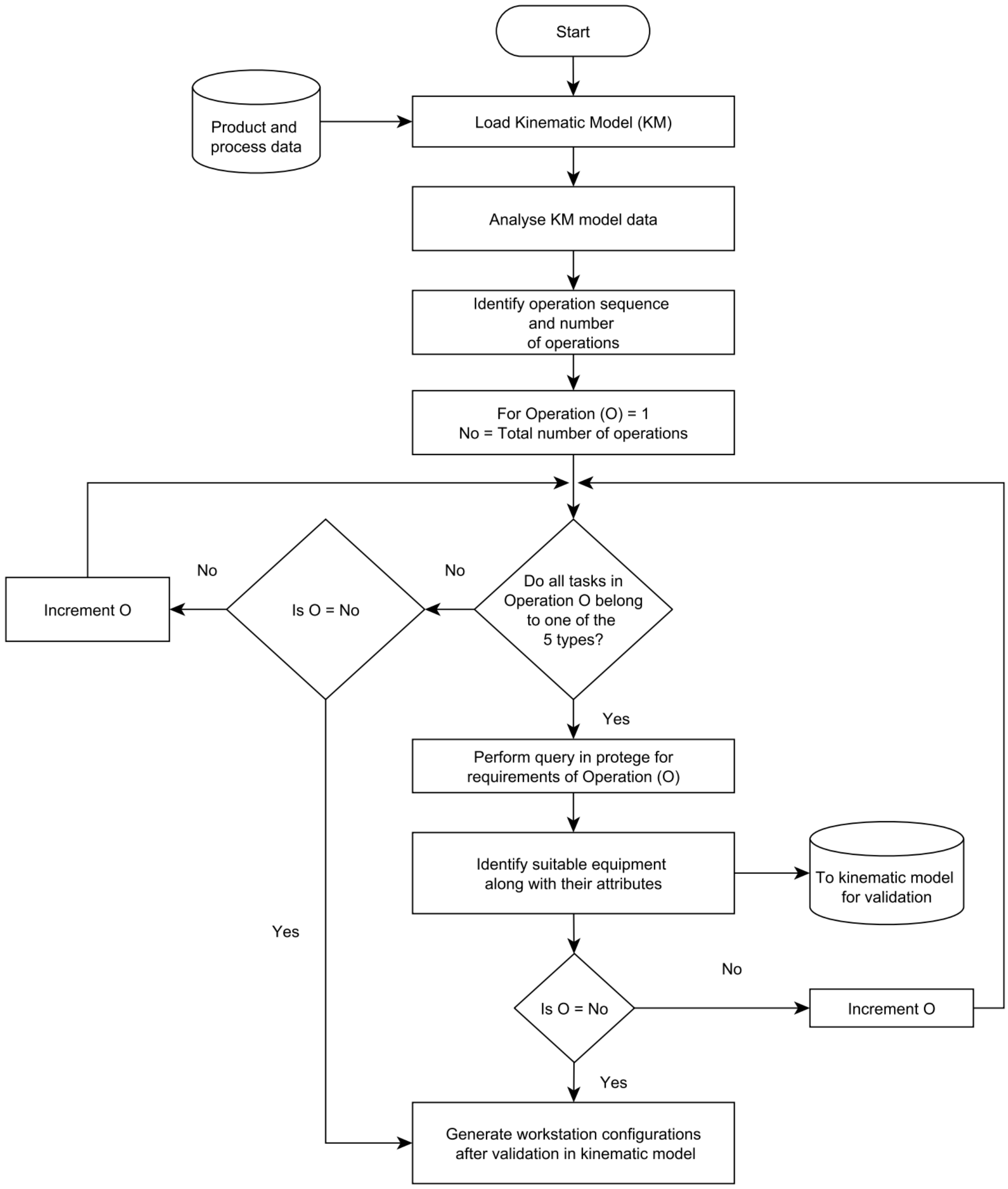


Figure 3.20: Protégé workflow.

Query design

In order to design the query, it is necessary to analyse the data obtained from the kinematic model. The operations are identified as *'O'* and the total number of operations is '*No*'. The information regarding the tasks performed in each operation are obtained from the kinematic model. The process of information retrieval is done with the help of the query language, *'Semantic Query-enhanced Web Rule*

Language' (SQWRL). In Protégé, primarily, three query languages are used and they are *'Simple Protocol And Resource description framework Query Language'* (SPARQL), *'Description Language'* (DL) and SQWRL. SPARQL is a standard Resource Description Framework (RDF) query language that inherently does not have understanding of Web Ontology Language (OWL). It is based on graph patterns and has a rich set of operators. It is widely available and not necessarily specific to Protégé. DL is a Protégé-specific query language and while it can be used for performing simple queries, it cannot do arithmetical operations and has a limited set of operators. SQWRL is a Protégé-specific language that understands OWL. Moreover, it is capable of doing arithmetical operations and it is a semantically robust, simple and expressive query language that is built upon the *'Semantic Web Rule Language rules'* (SWRL) [O'Connor and Das, 2009]. Conclusively, since the query that will be designed for the equipment selection for Stage one (WCS) will be a complex one with the involvement of arithmetic operators, SQWRL is chosen as the most suitable query language.

Using the query it is possible to screen a catalogue of equipment, that are defined as *'instances'* within component subclass of the resource class, to find those that are suitable to perform the required tasks as set out in the process sequence. To better illustrate the query process, a demonstration is provided in chapter 4, section 4.2.2. Since it is not possible to do certain validations that ascertain the feasibility of the solutions within the knowledge representation module, the selected equipment are first filtered within Protégé and then modelled in the kinematic module. For instance, after performing query, the selected equipment might meet all the required process parameters, but in reality, when it is installed in the workstation and begins operation, it might collide with an object in its path of motion. Although these discrepancies cannot be identified within the knowledge representation module, they can be diagnosed within the kinematic modelling module. In addition to validation using kinematic model, the workstation processing time can also be calculated, which is also highlighted in a previous work done by the author where the benefits of the integration of kinematic model and ontology module are discussed [Chinnathai et al., 2019]. The results of the query are stored in an XML format as shown in **Figure 3.21**.

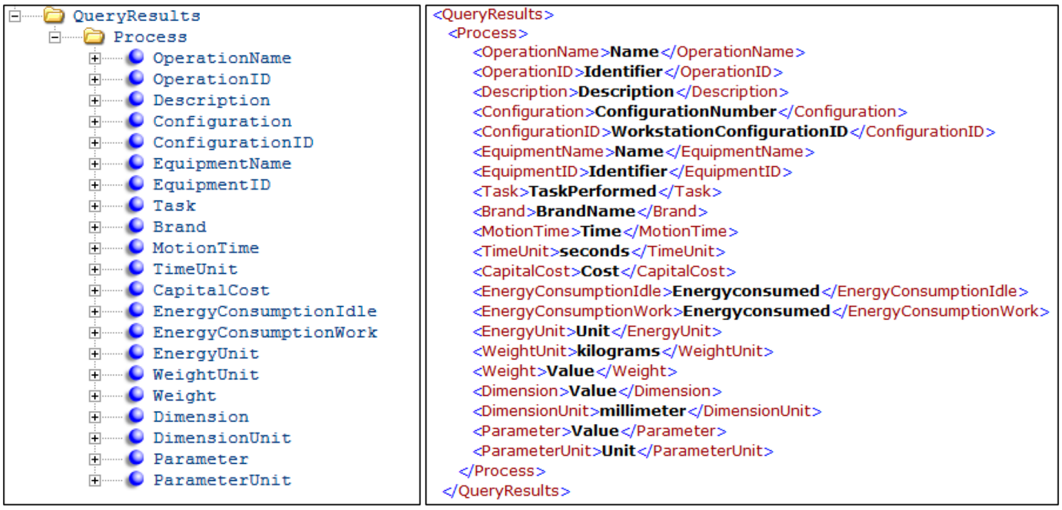


Figure 3.21: XML data structure for the query results from Protégé.

Summary of knowledge representation module

To summarise the characteristics of the knowledge representation module, it is necessary to explain how the kinematic model and ontology complement each other. The knowledge representation module is typically used to select those equipment that meet certain requirements or criteria and eliminate those that do not; the selection is done from a pool of standard off-the-shelf equipment that are available in the industry catalogue or equipment library. However, there are certain limitations in using this method. It is difficult to calculate the workstation process time, investigate collision detection, perform path planning and ergonomical analysis, and check assembly feasibility within the ontology editor. These issues can, however, be overcome by using the kinematic model and hence the coupling of both modules makes the virtual models more realistic. An important point to note is that the solutions provided at the end of the selection process in Protégé are by no means the only feasible solutions and there is always the possibility of designing bespoke equipment. Hence, the ontology-based selection process should be considered as an elementary guideline to support the equipment selection process.

3.3.3 Workstation design configuration and selection

The workstation design table comprises of the workstation KPI schema which serves as a template for the table. An example of the workstation design table is provided in chapter 4, section 4.2. The design table consists of the workstation candidate configurations that were validated in the kinematic model. In the design table, for each workstation configuration, the equipment that are suitable for the considered processes and the corresponding metrics such as investment cost, processing time, energy consumption, geometry and Computer-Aided Design (CAD) information of the workstation are highlighted. This data is important for the simulation optimisation that will be performed in stage 2 of the DDSM methodology. The production capacity of the post-scale-up facility depends on the process time; the cost values are important to ensure that the selected solutions are within the project budget and the geometry data is important for layout planning and ensuring that the selected workstation configurations can fit within the available space. The data is stored in the common database such that it is accessible by the software used in Stage two (SCS) of the methodology. This workstation design table also provides a good representation for system engineers to compare the alternate configurations before stepping into Stage two of the DDSM. **Figure 3.19** provides the XML data structure for the selected workstation configurations that will be stored in the common database.

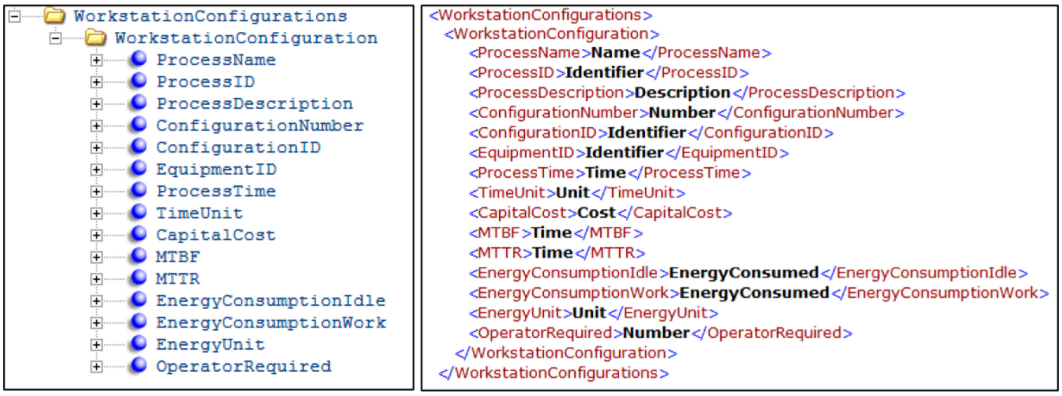


Figure 3.22: XML data structure for workstation configurations.

3.4 Stage two: System Configuration Selector

The primary aim of Stage two is to identify potential assembly line configurations with the help of two modules, *'DES Model'* module and *'optimisation'* module, that facilitate simulation optimisation for stochastic discrete-event systems. The benefits of using DES for production planning have been discussed in previous chapters; considering the wide spectrum of benefits provided by DES, it is not surprising that it is increasingly employed in the field of manufacturing for building models that allow the comparison of alternate scenarios, answering *'what-if'* questions and supporting decision making [Azab et al., 2012; Negahban and Smith, 2014; Jahangirian et al., 2010]. The inherent capability of DES to model production systems is a key reason to use it as part of the DDSM methodology. However, one major shortcoming in using DES in concept stage is that the system models in DES may be used to create scenarios that might, in reality, be impossible or impractical to build. To overcome this drawback, it is possible to integrate DES with kinematic modelling software such that the input data accuracy of the DES models can be increased [Chinnathai et al., 2019; Ghani et al., 2015; Ghani, 2013].

Optimisation is the process of finding one or more solutions that either maximise or minimise the formulated objective function while satisfying the defined constraints [Branke et al., 2008]. It is challenging to follow traditional optimisation approaches for stochastic systems due to the presence of probabilistic elements which make it difficult to derive a closed-form expression of the objective function. In such situations, it is possible to use DES to replace the closed-form expression of the objective function. Additionally, since real world complex manufacturing problems consist of a number of conflicting objectives, it is considered appropriate to employ multi-objective optimisation for the proposed research [Konak et al., 2006]. The software that will be used for performing the optimisation is MATLAB; the reason for choosing MATLAB is i) its capability to communicate to OPC-UA server using the OPC toolbox [MathWorks, 2021*a*], and ii) the availability of functions that support the visualisation of the optimisation progress for better control over the optimisation parameters [MathWorks, 2021*b*].

An overview of the workflow in Stage two is presented in **Figure 3.23**. For data communication between the DES module and optimisation module, real-time automated data integration is achieved with KEPServerEX using OPC-Unified Architecture communication protocol through which i) the optimisation variables such as the number of workstations, operators and material handling units are passed from MATLAB to FlexSim DES model and ii) variant-specific throughput is passed back

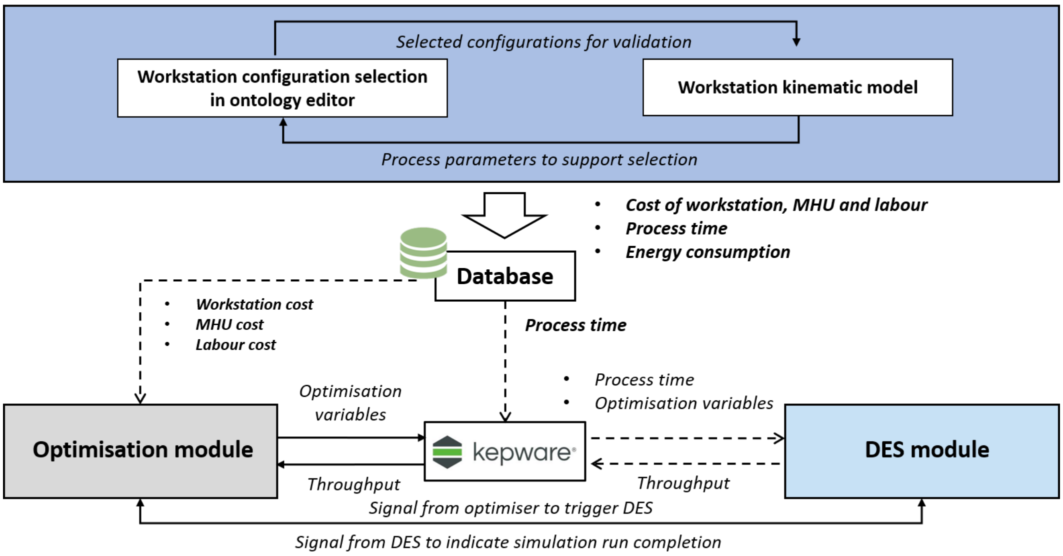


Figure 3.23: Stage two of DDSM methodology.

to MATLAB from FlexSim DES model. A more detailed description of the connection is provided in the following sections.

3.4.1 DES model module

Pilot line scale-up using DES to investigate the impact of scheduling policy and scale-up principles on certain system level KPIs was demonstrated by the author in a previously published work [Chinnathai et al., 2018]. As part of the research, additional stations and configuration changes were implemented in the DES model in order to meet the new demand. However, it was not possible to analyse the practicality of such solutions using DES alone. This was due to the top-down modelling approach using a standalone DES model in which there is no access to the workstation-level data. This is a major shortcoming that needs to be addressed since without the data from the workstation models, the process time, maintenance and energy values are assumed within DES models. Two important points to highlight about the DES module in DDSM is that i) it is the cornerstone for the assembly system configuration selection process, and ii) the modelling accuracy and transparency is improved by it receiving the data from the modules in Stage one. The workstation configuration data obtained from Stage one are represented as workstation KPIs and as explained in section 3.3.3, they are represented as XML schema and stored in the database. The conventional black box approach to the DES stations are challenged and the workstations in DES model are populated with data from lower level models. Additionally, certain user inputs such as the consideration of model abstraction, simulation graphic settings and parameters, representation of the process logic, etc., need to be provided for the model to perform good analyses.

Software selection

There are a number of commercially available software for DES modelling. For our purpose, the DES software that will be selected should be capable of communicating with other software and accept input decision variables from MATLAB and workstation KPI information from the lower level models. Moreover, it should be capable of modelling human operators, performing energy consumption analysis and breakdown analysis depending on the application. It should also be capable of modelling the randomness of a manufacturing system. Considering scale-up in specific, since the material handling units and their activities play a major role in increasing the productivity of the system, importance should be given to the modelling of material handling units/transporters. Additionally, there is need to run multiple replications for each combination of parameters that are decided by MATLAB. For this purpose, the DES software should have the capability to perform replications with random streams. Considering all these requirements, FlexSim was selected as the most suitable DES software, primarily due to its availability and relative ease of communication with OPC-UA servers. Additionally, the availability of pre-defined library elements within FlexSim, that represent the production system is very beneficial for modelling in the planning phase. Moreover, the FlexSim user community is quite active and it is possible to find and utilise the user-defined libraries that are posted in the official forums. The following paragraphs provide more details about FlexSim DES software and why it is considered suitable for the DDSM framework.

FlexSim: Overview

FlexSim is a DES software that is primarily used for manufacturing, healthcare, material handling and warehousing simulations. The software has options to use the drag and drop method of modelling or the process flow-based modelling or an integration of both to cater to the user's needs. The FlexSim library consists of i) fixed resources such as source, queue, sink, etc., ii) task executors such as operators, transporters, robots, ASRS, iii) travel networks, and iv) conveyors. Additionally, detailed modelling of AGV, algorithm for AGV path planning, fluid modelling are also available. This kind of support in modelling material handling units is very useful for the scale-up analysis. Moreover, the FlexSim toolbox consists of breakdown modelling, dashboards, process flow elements for flow chart, statistics tracker, connectivity tool, shift tool and global tables.

FlexSim: Operator and transporter modelling

FlexSim comprises of a model element called *'dispatcher'* which allows centralised control of operators and transporters in the DES model. The dispatcher is responsible for selecting which operator and in what sequence they will perform the assembly operations. The operators can either follow a pre-defined path or they can define a new path based on the start and destination positions. The dispatcher and operator modelling in FlexSim is presented in **Figure 3.24**. Similarly, the transporter modelling also involves the use of dispatchers that are used to select and schedule the activities of the material handling units. The parameters used for transporter modelling varies depending on the type of material handling unit used. Various elements such as AGVs, vehicles, robot and operators can be used for the purpose of material handling and in general, the relevant parameters such as the travel speed, acceleration, collision spheres for collision detection, queue strategy and priority need to be added and many of these can be defined as distributions. Similar to the operator path planning, for the transporter travel path, pre-defined networks can be used or the inherent algorithms for path planning can be used. The transporter modelling in FlexSim is provided in **Figure 3.25**

FlexSim: Workstation modelling

In FlexSim, the workstations are referred to as *'processors'* and they have certain properties that can be defined using the interface. These include defining the

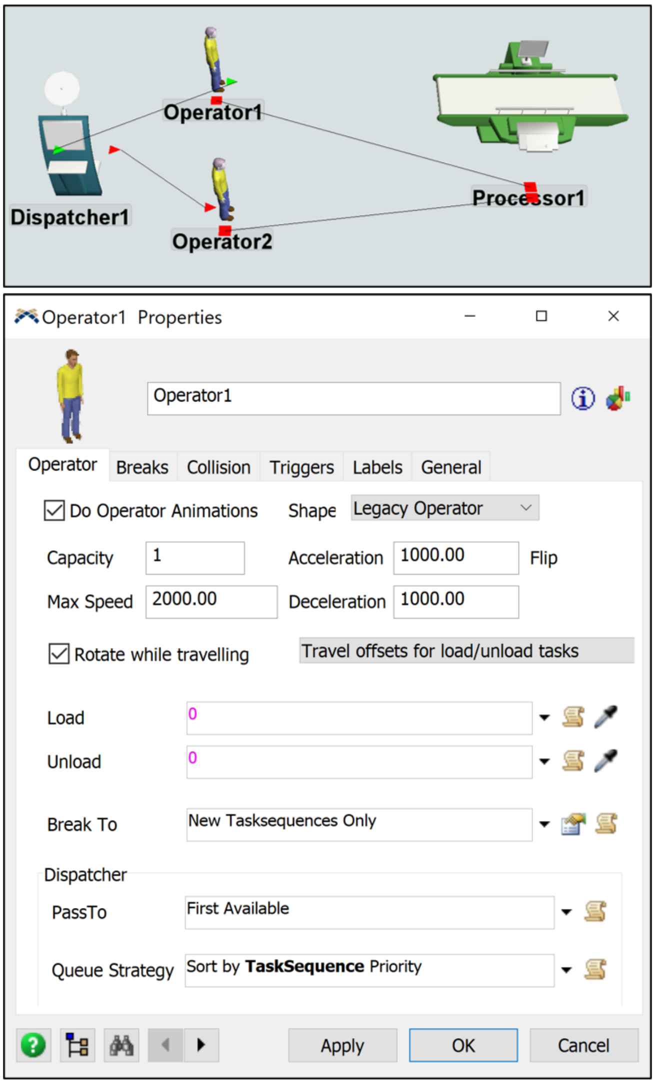


Figure 3.24: Operator modelling in FlexSim.

setup time, process time, input and output ports, operator or transporter requirement, etc. The input and output ports are used to determine the flow of workpiece through the system. The processors also have the capability to perform certain actions when workpiece-specific events such as *'workpiece entry', 'workpiece processing', 'process finish', etc.*, take place. The actions include updating the variable or label values, changing workpiece dimensions or colour, reading label values of the workpiece, etc. With these actions it is possible to enrich both the visual and statistical aspects of the simulation. An image of the workstation modelling in FlexSim is provided in **Figure 3.26**.

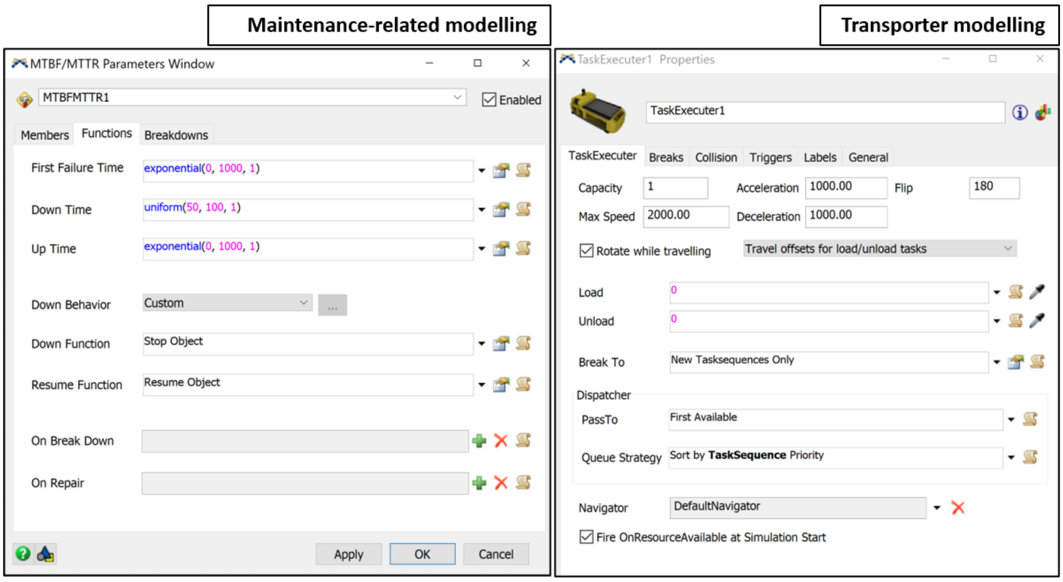


Figure 3.25: Maintenance and transporter modelling in FlexSim.

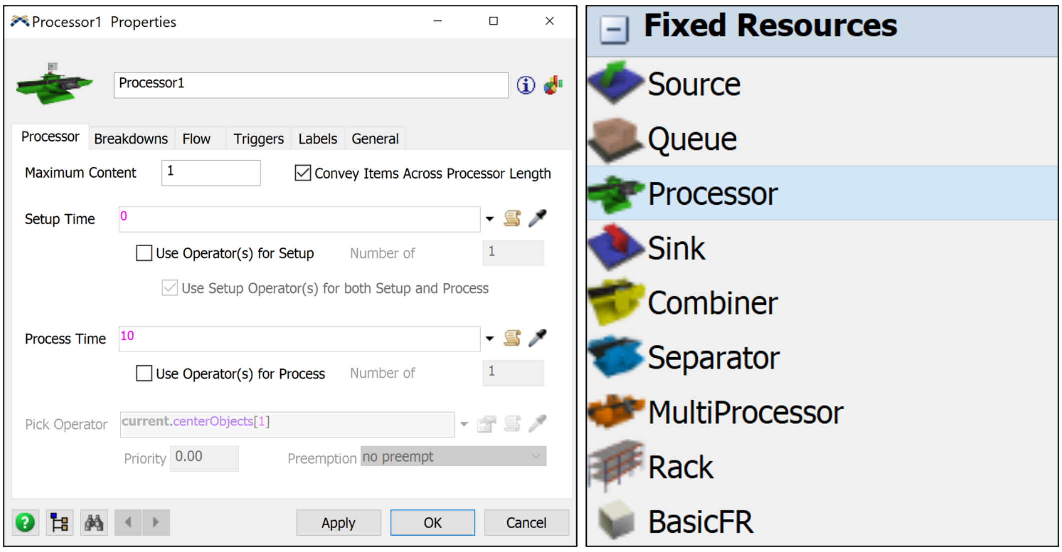


Figure 3.26: Workstation modelling in FlexSim.

FlexSim: Maintenance modelling

It is also possible to do maintenance-related analysis within FlexSim using the MTBF/MTTR tool. The FlexSim maintenance modelling user interface is provided in **Figure 3.25**. The operators, workstations and transporters are added as members to the created MTBF/MTTR element and the functions such as first failure time,

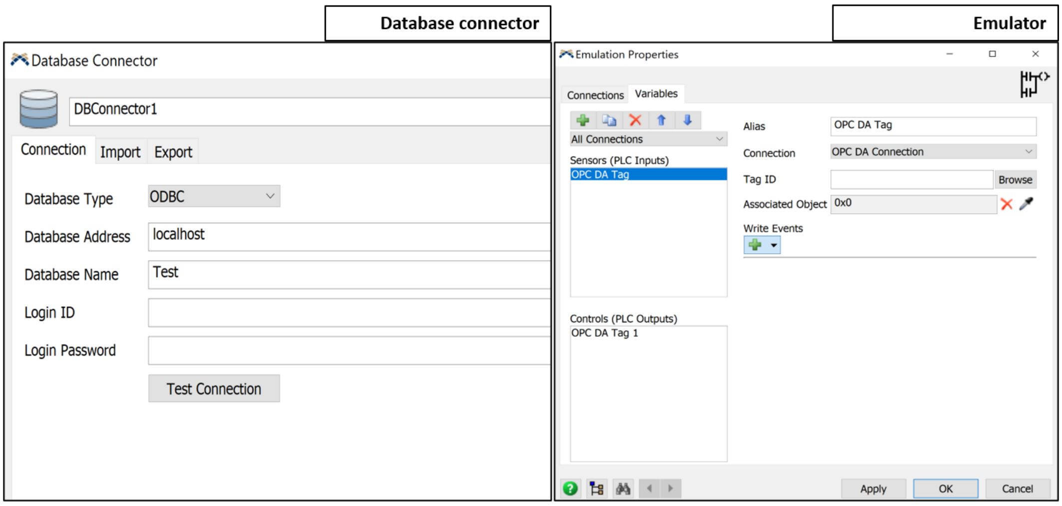


Figure 3.27: Connectivity tool in FlexSim.

down time and up time are defined in the form of probability distributions. Typical scenarios that can be considered as breakdown include the re-calibration, tool wear, breakage and stoppages due to programming error. When considering transporters, the breakdown scenarios include the unprecedented stoppages, battery recharge or fuel refill and waiting for parts. In case of operators, the breakdown includes taking break outside rest hours, waiting for parts, etc. In addition to the maintenance modelling described above, it is also possible to consider scheduled maintenance and predictive maintenance.

FlexSim: Connectivity tool and experimenter

The core feature that makes FlexSim suitable for the DDSM is the connectivity tool that enables integration with other software. The connector enables communication to the database using the database connector, and the OPC UA server using the emulator. **Figure 3.27** shows the database connector and emulator in FlexSim. Within the emulator, necessary variables are created and depending on whether the values need to be imported or exported, they are defined as *'read'* variables or *'write'* variables. For each tag that will be created in the KEPServerEX, the corresponding variable should be created within the emulator. Another important feature of FlexSim that is essential for simulation optimisation is the experimenter. **Figure 3.28** shows the experimenter tool in FlexSim. For each iteration of the

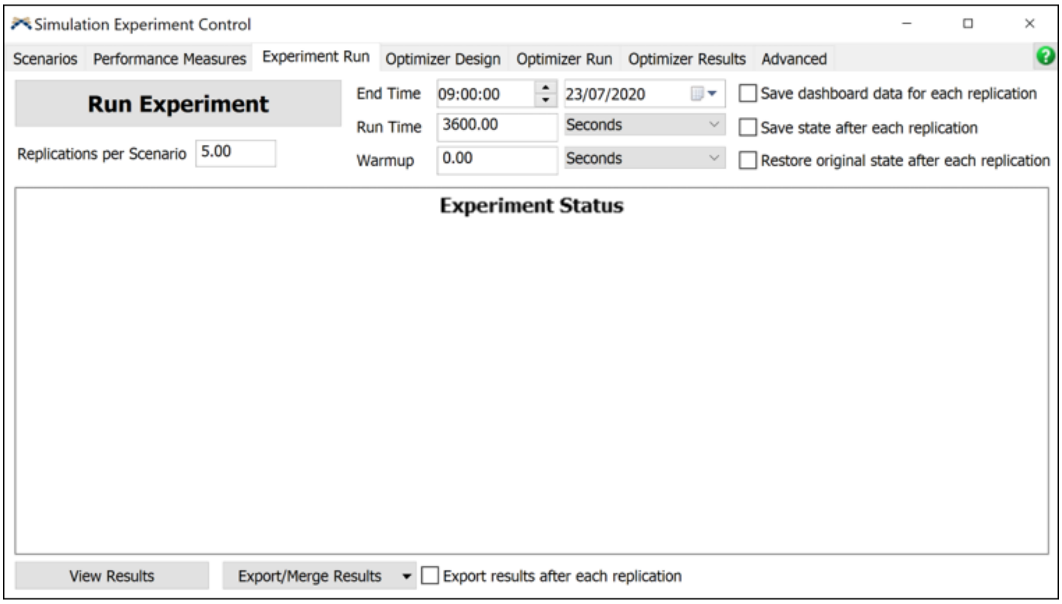


Figure 3.28: Experimenter tool in FlexSim.

optimisation, several runs of the simulation need to be done to obtain a meaningful result. Within the experimenter, several scenarios can be defined along with the number of replications for each scenario. The performance measure that needs to be evaluated is also defined and the parameters for the experiment such as the run time and warm-up time are provided. The last important feature that is essential for simulation optimisation is the ability to trigger the DES software and control it from outside the DES platform. In the case of FlexSim, it is possible to trigger and control the model start, run speed, etc., by using a *'batch file'*. This enables triggering the software automatically each time a simulation run is needed and an example script is provided in **Figure 3.29**.

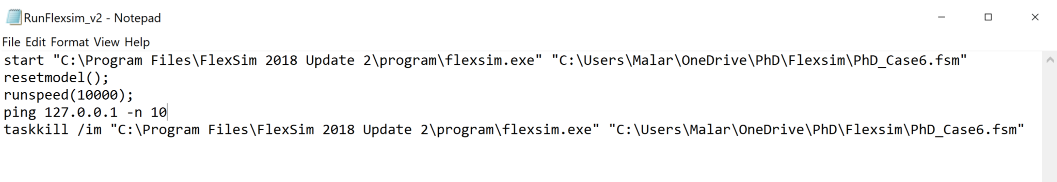


Figure 3.29: An example of batch file for automating simulation optimisation.

Parametric DES model

Creating a parametric DES model allows it to reach the desired level of configurability and adaptability which is considered vital for simulation optimisation [Aggogeri et al., 2015]. The parameters that are created can take up different values depending on the optimisation module. An example of a parametric DES model is provided in **Figure 3.30**. From the Figure, the parameters in the DES include the quantities of workstation, operator and AGV. The actual values of these parameters are not known and hence they are registered as variables. The values are then passed from an external software such as the optimisation module to the DES model.

In DDSM, the considered parameters include the quantity of operators, AGVs, workstations and type/configuration of workstations. The reason for choosing these parameters is that they are the decision variables for the optimisation procedure and this will be explained in detail in section 3.5. The next step is to ensure that these parameters can be modified from outside the DES model automatically. To achieve this, the connectivity tool is utilised and the variables are defined such that they can communicate to the KEPServerEX. Any changes to the values in this server will automatically update the parameters within DES on simulation start. This method can be applied to create a generic model that can be adaptable to different scenarios.

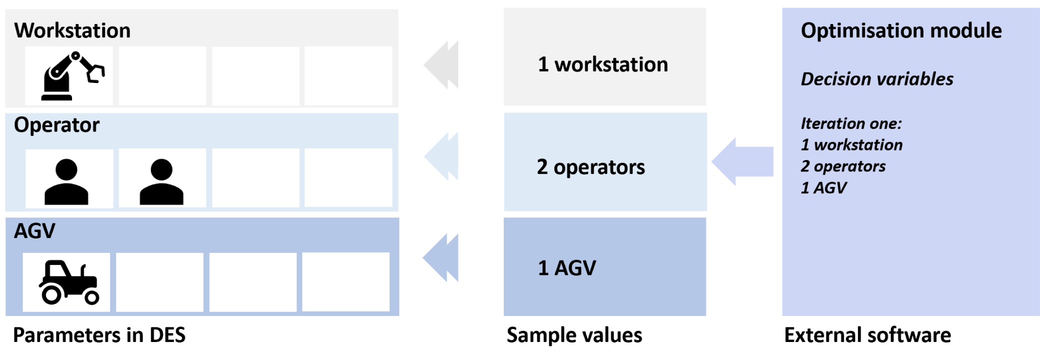


Figure 3.30: Illustration of a parametric DES model.

The DES module comprises of the FlexSim software and its characteristics and capabilities were discussed. The procedure to create a parametric DES model was also provided. Depending on the test case, the relevant layout needs to be considered along with other DES-specific data and the model needs to be created. On creation of the model, the parameters need to be identified and the connection with server needs to be established. The next step is to verify and validate the model to ensure that the model is performing what it was intended to. Additionally, it is important to have certain number of test runs to check the connectivity with the server. The verification and validation phase can go through some iterations until the designer is confident that the model will produce reasonable results. The next step is to explain the working of the optimisation module.

3.4.2 Optimisation module

In this section, the fundamentals of the optimisation module, the connectivity and data transfer are discussed. The actual problem formulation and Genetic Algorithm (GA) will be discussed in the section 3.5. The optimisation module is responsible for optimising the model using GA and is coded in MATLAB. The reason for choosing MATLAB being the availability of software and the existence of extensive functions that allow the optimisation without the need to do a lot of coding. The reason for choosing GA will be explained in the following paragraphs.

Multi-objective optimisation

Multi-objective optimisation focusses on maximising the rewards or minimising the costs. It involves the optimisation of more than one objective function which may be conflicting [Wang et al., 2011]. The problems that involve finding the value of the decision variables or parameters are referred to as parametric optimisation problems [Gosavi, 2015]. Since real world problems are not simple enough to be represented using one objective alone, it would be artificial to try to reduce the number of objectives; it is also not realistic to aggregate them.

In multi-objective optimisation, there are three phases: **model building, optimisation and decision making**. During the model building, the optimisation objectives and decision variables are set out. The decision variables can take up values between a certain range and can either be discrete or continuous. For each iteration of the optimisation, the values of the decision variables change depending on the algorithm considered. These values are referred to as the decision variable vector. At the end of the optimisation iteration, the scores of each decision variable vector corresponding to the considered objectives are evaluated. The decision variables that have relatively better score values are then plotted on the pareto-front that represents the value of the objective function one vs. objective function two in case of bi-objective optimisation.

The decision making generally involves the preferences of the decision maker with the option of incorporating his preferences before the actual run of optimisation or post-optimisation [Branke et al., 2008]. There are a number of different algorithms available for performing multi-objective optimisation and evolutionary algorithms represent one such category. The evolutionary algorithms use a population-based approach that allows finding multiple solutions that are non-dominated, simultaneously for a single iteration of the optimisation. In general, evolutionary algorithms aim to find a set of pareto-optimal decision vectors that are diverse enough to represent entire range of the non-dominated solutions [Wang et al., 2011].

Multi-objective optimisation allows better understanding of the problem and available alternatives and ultimately helps make better choices. When the considered objectives are conflicting, the resultant solutions represent a trade-off between both objectives considered, which is represented in the form of pareto front. Analysis of pareto front helps to gain a better understanding of the inter-dependencies among the decision variables, objectives and constraints [Branke et al., 2008].

Simulation-based multi-objective optimisation

Simulation-based multi-objective optimisation follows the same procedure of a multi-objective optimisation, where an objective function is defined during model formulation along with the consideration of decision variables and their range of values. The key difference is the coupling of simulation model to the optimisation algorithm. For modeling the complex manufacturing problems, it is difficult to obtain a closed form of the objective function due to the presence of probabilistic elements such as probability density function or cumulative distribution function for one or more variables in the objective function. Since the closed form of objective function is time consuming to obtain and difficult to calculate, an estimation can be obtained from simulation.

For example, consider an optimisation problem where the objective is to maximise the daily system throughput. In this case, it is challenging to represent this objective function as a closed form as the throughput is the culmination of the interaction between a number of elements in the production system. Therefore, in such situation, DES can be used to obtain the values of the throughput to provide an estimate of the objective function. The more the number of samples obtained from DES, the closer the estimate to the actual value of the objective function. It is in these situations, simulation plays a significant role to estimate the value of the objective function in the optimisation procedure. For this purpose, the optimisation techniques that do not require the closed form but only the numerical values of the objective functions are paired with DES to perform the simulation optimisation procedure [Gosavi, 2015].

Parametric numerical optimisation techniques are suitable for simulation optimisation and these techniques are also called as model-free or black box techniques and the underlying assumption is that it is possible to obtain the true value of the objective function when averaging the objective function values over numerous simulation replications for a decision variable vector in the design space. The vector, in design space, indicates a set of values for the decision variable that are considered for the optimisation.

The use of exhaustive simulation optimisation techniques is very tedious and time-consuming to the extent that it makes it infeasible. The alternative is to use *'meta-heuristics'* that instead of scanning the entire design space, follow heuristics to search for good pareto-optimal solutions. Although meta-heuristics do not have the best of convergence properties, they are useful for complex manufacturing problems where the consideration of a number of decision variables makes it infeasible to explore the entire design space to arrive at optimal solutions. Although meta-heuristics do not guarantee optimal solutions they provide good solutions and the oldest of the meta-heuristics is the *'Genetic Algorithm'* (GA); it has been extensively used in industries with success and it is compatible with DES for simulation optimisation. It is an evolutionary algorithm that involves the selection of popula-

tion members and computation for each member of population space; they are then sorted according to the domination principle [Goldberg, 1989].
In GA, the individuals of each generation are comprised of different values of the decision variables and a certain number of these individuals make up the population. Each simulation run corresponds to one individual from the population selected and their decision variable values are used to control the simulation parameters. Through the process of evolution, fitter solutions are selected for subsequent generations. Two essential operators, *'mutation and crossover'* are used to generate new solutions. During crossover, a portion of population participate in the crossover and the remaining are directly taken along with the child populations. The child solutions are subjected to mutation depending on some predefined probability values [Goldberg, 1989]. Crossover operator is considered to support convergence by combining two chromosomes of parents to form new chromosomes. In such a way, it is expected that good chromosomes appear more frequently. Mutation introduces diversity back into the population and is vital for escaping the *local minima* [Konak et al., 2006].

Steps in optimisation module

In DDSM, the optimisation module focuses on multi-objective optimisation and considers two conflicting objectives. For this purpose, *'gamultiobj solver'*, which is a MATLAB solver for optimising multi-objective problems is used. The solver is built upon a controlled elitist GA which is a form of Non-dominated Sorting Genetic Algorithm (NSGA-II) [Deb et al., 2003] that uses the *'elitist principle'* to preserve diversity and emphasise the non-dominated solutions; the diversity of the set of non-dominated resultant solutions is considered essential for convergence. Elitism operators aim at keeping the better solutions from the combined old and new populations to ensure a performance that cannot degrade. Elitism is controlled by the *'pareto fraction'* and *'distance function'* options. The former limits the number of solutions on the front and the latter favours diversity.

The pareto front plays a key role in decision making and represents the solutions that exhibit a good trade-off for both considered objectives. **Figure 3.31** shows a representation of the pareto front. The pareto front population fraction determines

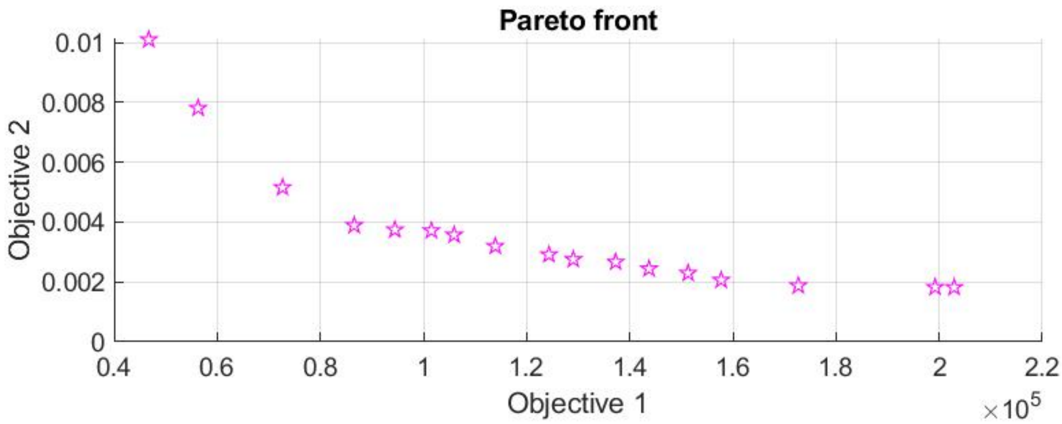


Figure 3.31: Example of a pareto chart obtained from MATLAB.

the number of solution points on the pareto front; the default value for population fraction in MATLAB optimisation toolbox is 0.35. It is important to ensure that the individuals represented on the pareto front are diverse enough to represent the range of pareto front. The pareto front solutions around the *'curved part'* of the front exhibit acceptable fitness scores for both the objectives considered. The initial set of population is selected at random and subsequent populations for future generations are chosen using non-dominated rank and distance function. The individuals are given a non-dominated rank depending on their fitness value. The distance function, namely '*crowding distance*', is used for selection when two individuals of a population have the same rank. Typically, three different stopping criteria can be considered for termination of the optimisation. These are: *i*) maximum number of generations for which the optimisation will run, *ii*) the stall generation limit which checks for optimisation convergence using a tolerance value, and *iii*) maximum time limit for the optimisation run. The steps involved in the simulation optimisation given in **Table 3.2**.

3.4.3 Data exchange between DES and optimisation module

Two sets of data will be exchanged between the DES module and optimisation module. As explained in **Figure 3.10**, the dataset (f) represents the data that needs to be communicated to DES module from optimisation module and the dataset (g) represents the data that needs to be communicated to the optimisation module from

Table 3.2: Genetic Algorithm pseudo code.

Pseudo code of the GA	
(1)	Initialisation and population selection;
(2)	Evaluate the initial population through fitness function;
(3)	**For** (generation < max gen.)
(4)	**While** (not meet the stopping criteria)
(5)	Select parents for next generation using binary tournament selection;
(6)	Create children using mutation and crossover;
(7)	Combine current population and children;
(8)	Compute rank and crowding distance;
(9)	Trim population size;
(9)	**End While**
(10)	Evaluate the new population fitness;
(11)	**End For**
(12)	Output the best solutions;

the DES module. As part of dataset (f), the values of the decision variables that are decided within the optimisation algorithm in MATLAB need to be passed on to FlexSim. Each decision variable is associated with a corresponding variable inside FlexSim and it represents a particular parameter. The communication is achieved in real-time with the help of OPC UA communication protocol and KEPServerEX software. Depending on the number of decision variables considered, tags are created within KEPServerEX to store the values of the variable. The communication between MATLAB and the server is established using *'OPC toolbox'* in MATLAB. On the other hand, FlexSim is also connected to the server using the *'emulator tool'*. The tags in the server hold the values of the decision variables which are passed to FlexSim. Similarly, at the end of the simulation run in FlexSim, the throughput values are passed back to MATLAB using the server. Using this approach, the datasets (g) and (f) are communicated across both modules. Depending on the software used for the DES and optimisation, this procedure might vary. However, the underlying approach and objectives remain similar. The generic workflow of the data exchange between the MATLAB and FlexSim is explained in the following steps. A detailed explanation with case study is provided in the next chapter.

1. The initial set of values for decision variables are decided in MATLAB *'initial population'* function randomly and the first iteration is now initialised.

2. In the first iteration, the values of the first member of the population, which is essentially a combination of decision variable values, are passed from MATLAB to FlexSim through the server along with the signal to trigger FlexSim.

3. The simulation model is run for the pre-defined parameters of speed, warm-up time and simulation run time for a certain number of replications.

4. The average throughput value for the considered product variants are calculated at the end of the simulation run and passed back to MATLAB through the server.

5. As the simulation terminates, a signal is passed back to MATLAB to continue the optimisation process, such that the obtained throughput values can be used to calculate the objective function.

6. In this way, the optimisation process continues for the next member in the population till all the members are evaluated; this constitutes one generation. The next generation is initialised and the process continues until the stopping conditions are met.

3.4.4 Decision making module

On meeting the stopping conditions, the pareto front can be obtained by plotting the objective scores of the best members of the population over a number of generations. The scores represent how well the values of a particular population member is good at either maximising or minimising the value of the objective function. The results from the pareto front will be analysed in the decision making module.

The decision making module obtains input from three different sources, *'the user priorities, optimisation module and scale-up KPI schema'*. The scale-up KPI schema refers to a set of criteria that are used to compare the pareto front solutions obtained from optimisation. The user priorities refer to the preferences of the decision maker when it comes to selecting a solution after comparison. The pareto front is obtained from the optimisation module. To enable the decision making, four main criteria are considered; they are throughput, compactness, ease of transition and cost-efficiency.

The *'compactness'* is a measure of the space occupied and is represented as c2/slots occupied. The number of slots occupied is obtained from the total number of workstations that are employed for that particular solution and the ease of transition is a subjective criteria that is obtained from discussions with system engineers and process planners. Since the compactness is the inverse of the space occupancy, the constant c2 is introduced. The *'cost efficiency'* is represented as c1/scale-up cost, and since the cost efficiency is the inverse of the scale-up cost, the constant c1 is introduced. The scale-up cost values are calculated using the objective function. The *'throughput'* is considered as the total number of each product variant that gets processed in the assembly system and received at any designated buffer or sink after the completion of assembly at the last workstation on simulation run completion. The throughput values are obtained from FlexSim.

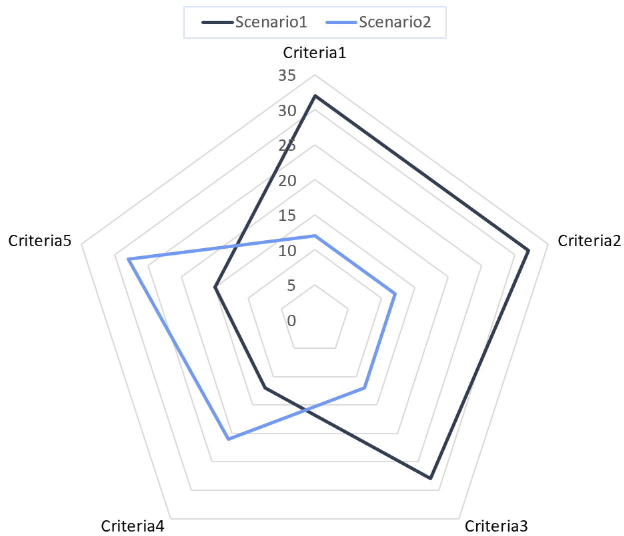


Figure 3.32: Sample radar chart representation of the alternate design solution.

The visualisation of the results is done with a radar chart as shown in **Figure 3.32**, where the higher the value of a solution for a particular criteria the better that solution satisfies that criteria. Since the ultimate aim is to compare the solutions obtained from the optimisation model, the solutions should be judged across a common ground and hence the total simulation time is kept as a constant across all the simulation runs. It is assumed that infinite number of products are available for processing to avoid starvation of workstations that might artificially reduce the efficiency of a solution. On the downside, it is difficult to compare the solutions with regards to scheduling policies that actually prioritise one product over the other. For such cases and applications, it might be better to allocate a finite number of products at the start of the simulation and consider the simulation time as a variable.

This section discussed the Stage two of the DDSM methodology which is the System Configuration Selector (SCS). The DES module, optimisation module and the data exchange between these two modules was discussed. The pareto front and the decision making from the results of the simulation optimisation was also discussed as part of this chapter. The next section explains the problem formulation and objective function definition in more detail.

3.5 Problem formulation

3.5.1 Objective functions

In this section, the objective function that will be used in the optimisation module is explained in detail. The considered optimisation problem has two conflicting objectives, *i*) *'scale-up cost'* which is detailed in **Equation 3.1** and *ii*) *'system throughput'* which is detailed in **Equation 3.2**. The specific aims of this optimisation study are to: *i*) identify the number of workstations of each type required, *ii*) identify the number of operators of each type required, *iii*) identify the number of Material Handling Units or transporters of each type required and iv) identify the suitable configuration of workstations such that the required throughput can be achieved while within the scale-up budget. All considered mathematical notations are given in **Table 3.3**.

The workstations are categorised according to the operations performed and all workstations that perform the same operation belong to the same type (represented as '*w*'). For some of the considered workstation types, different alternative equipment performing the same operation are identified in Stage one to obtain the different workstation configurations. To put this into perspective, consider an example where two assembly operations, pick and place and welding are done. Operation one and two require a workstation each and these two workstations differ in their type. Therefore, *'w = 1'* and *'w = 2'* represent the workstations that perform operation one and two, respectively; they are referred to as workstation types one and two. For workstation type one, there can be different workstation configurations, as per the analysis in WCS of DDSM the methodology. The workstation type one is associated with a variable, within the optimisation module, to determine which of these alternate configurations will be selected for a particular iteration. Similarly, variables are used to identify each workstation type that has alternate configurations. There may be some workstation types for which only one configuration is considered. In such situations, there is no configuration variable associated with that workstation type. In this way, the workstation types with alternate configurations are associated with a decision variable; this variable corresponds to the workstation configuration chosen for a particular iteration of the simulation optimisation.

Four types of decision variables are considered for the optimisation study as follows:

- x_i^1 $(i = 1, ..., N_w)$ to decide the number of each type of workstation required,
- x_j^2 $(j = 1, ..., N_m)$ to decide the number of each type of MHU required,
- x_k^3 $(k = 1, ..., N_o)$ to decide the number of each type of operator required,
- x_l^4 $(l = 1, ..., N_t)$ to decide the workstation configuration of each type of workstation considered.

The first decision variable is used to indicate the number of copies of a particular workstation type that is considered. Along with the second and third decision variable types, the three initial variable types are related to the replication principle of scale-up and the fourth decision variable is related to the principle of upgrading. It is for the fourth decision variable that the input from Stage one regarding the workstation configurations are necessary. The other three variable do not need input from Stage one. Additionally, two types of design constraints are considered in this case study: *i*) integer constraints and *ii*) bound constraints. The integer constraints are defined to allow GA to perform the optimisation for integer decision variables. The bound constraints are used to limit the maximum number of stations, operators and transporters due to budget restrictions.

Table 3.3: Notations.

Notation	Description
w	index to represent the workstation type
NW	total types of workstations
K_w	total number of workstations of type 'w'
t	index to represent the workstation types that have alternate configurations
N_t	total number of workstation types with alternate configurations
S_w	cost of workstation of type 'w'
m	index to represent the MHU type
NM	total types of MHU
Q_m	total number of MHUs of type 'm'
M_m	cost of MHU of type 'm'
ω	index to represent operator type
NL	total types of operators
R_ω	total number of operators of type 'ω'
W_ω	hourly wage of operator type 'ω'
T	total production time in hours
T_ω	total shift time for operator type 'ω'
β	penalty cost for exceeding the available space
p	index to represent product type
N_p	total number of product variants
ε_p	throughput of product 'p' at the end of time 'T'

Objective 1 is the scale-up cost which consists of four main elements. The first element is the investment cost of adding new machines, the second element is the cost of material handling units and the third element is the cost of labour. The fourth element is a penalty cost for exceeding the available space; the space is represented

as slots within which workstations can be added. If the space restrictions are not violated, then the penalty cost, β, is zero. However, on violation of the space constraint, the penalty cost is a number that is five times the summation of other cost elements. The method of determining the penalty cost can be changed but the important point is to ensure that the chosen value penalises the solution that exceeds the space constraint. This way that particular solution will not be considered for the next generation.

$$\begin{aligned} f_1(x_i^1, x_j^2, x_k^3, x_l^4) = Min(\sum_{w=1}^{NW}(S_w \cdot K_w) + \sum_{m=1}^{NM}(M_m \cdot Q_m) \\ + \sum_{\omega=1}^{NL}(W_\omega \cdot T_\omega \cdot R_\omega) + \beta) \end{aligned} \tag{3.1}$$

Please note that, the direct and indirect raw material costs, indirect labour costs and maintenance costs are not considered in this objective function. The reason is that their impact on the decision making process is considered less important than the considered scale-up cost elements. However, the scale-up cost function can be modified depending on the application and preferences of the decision maker.

Objective 2 is to maximise the system throughput. In this case, all product variants are assumed to be equally important and hence no weights are considered. However, if there arises a scenario where product have different priorities, then the equation should be adjusted accordingly. The total system throughput is considered to be the sum of all product variants that are assembled and collected at the last workstation or collection point at the end of the production time T.

$$f_2(x_i^1, x_j^2, x_k^3, x_l^4) = Max(\sum_{p=1}^{N_p} \varepsilon_p) \tag{3.2}$$

3.5.2 Assumptions

The following assumptions are considered for the optimisation. The station footprints of all workstations are assumed to be the same. The production facility is divided into a number of slots to represent the available space and each workstation occupies only one slot. The new demand for which the scale-up transition is done is assumed to remain constant during the simulation period. Only production line and associated operations are considered and it is considered sufficient to perform the analysis required for the DDSM methodology.

3.6 Summary

The primary aim of this chapter was to introduce the two-stage methodology that supports data integration at two different levels of granularity, the workstation level and system level and explore and identify the data and their sources that are necessary to support scale-up decision making. Additionally, the data format and formulation of the optimisation problem were explained with relevant examples, tables and figures. The novel concept of using data from a knowledge-based kinematic model to support the simulation optimisation process was introduced and the workflow of the knowledge mapping and kinematic model creation was presented. This chapter also provided an approach to achieve the data integration between the kinematic and DES models to support assembly system decision making. This brings the readers to the next chapter which will discuss the implementation of the methodology on a battery module assembly line.

Chapter 4

Case study

A detailed explanation of the two stage methodology was provided in the previous chapter. Moving on chapter 4, the DDSM methodology will be applied to a battery module assembly line to demonstrate its use and benefits. The chapter starts with an explanation of the battery module assembly line and then demonstrates the application of the methodology for the considered case and concludes by discussing the pros and cons of the DDSM methodology.

4.1 Case study overview

The considered case study is a prototype battery module assembly facility that is located in Warwick Manufacturing Group (WMG) at the University of Warwick. The pilot line is part of a project named AMPLiFII - Automated Module to pack Pilot Line For Industrial Innovation. An image of the pilot line facility at the University of Warwick is presented in **Figure 4.1**. The same facility is modelled in VueOne and represented in **Figure 4.2**. The vision around the assembly line is the use of a modular assembly system design that can cater to the needs of different battery module designs that comprise of cylindrical battery cells. Typical assembly processes include battery testing, welding, module testing and cooling system assembly. The scale of production is very small in the range of 2 - 5 modules per day since it is a pilot line where industrial partners are able test their processes and products. However, once the products and the respective processes are tested, the industrial partners have plans to build a large scale battery assembly facility for which the pilot line is considered as the representative model. To expand this pilot

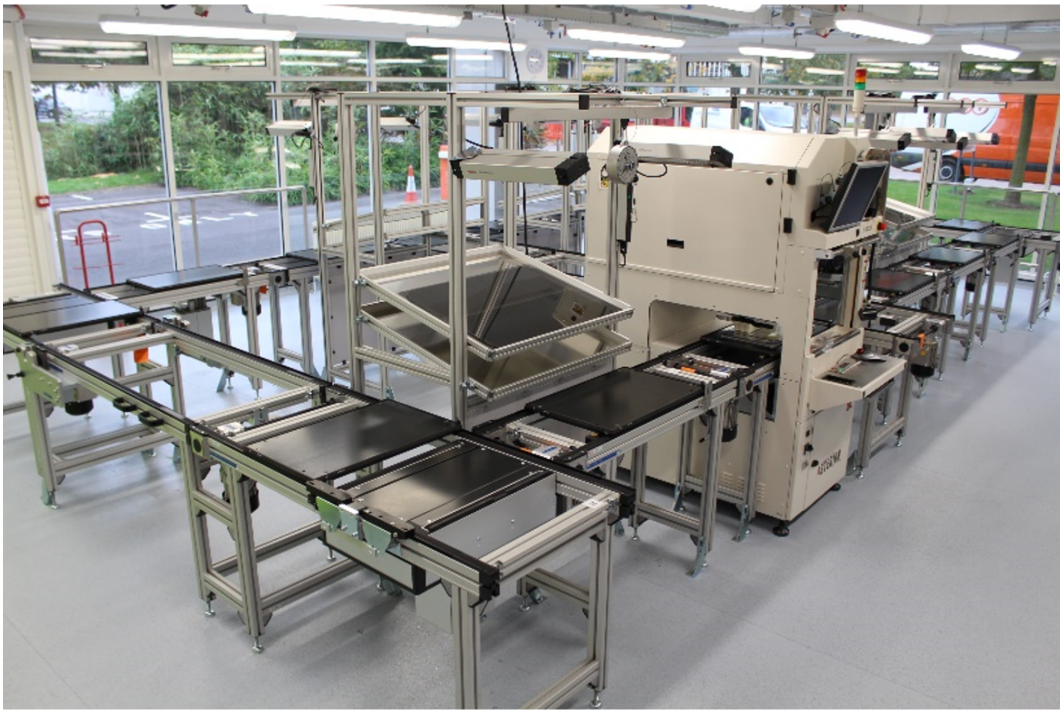


Figure 4.1: Pilot line for battery module assembly in WMG.

line design into a fully operational assembly line is a challenging task since the approach towards the up-scaling of the pilot level design to a high-volume production facility is not clearly understood due to lack of experience and novelty of the electric vehicle powertrain and cylindrical lithium ion battery technologies. Especially with regards to the electric vehicle battery manufacturing and assembly, there are various problems when moving from the concept phase to full-scale production that stem from the material handling, labour and material feeding, functionality change and automation.

Although several different product designs have been tested in the pilot line, two different variants of battery modules, module A and B, are considered for this case study. The two variants differ in their designs and the number of cylindrical cells they house. The assembly process for both variants is similar with some variations present in the welding and cooling system assembly. A total of five assembly workstations assemble the battery modules and each workstation is virtually modelled with the details of the geometry, kinematics and logical behaviour in the kinematic modelling software. The assembly operations are allocated to the workstations as

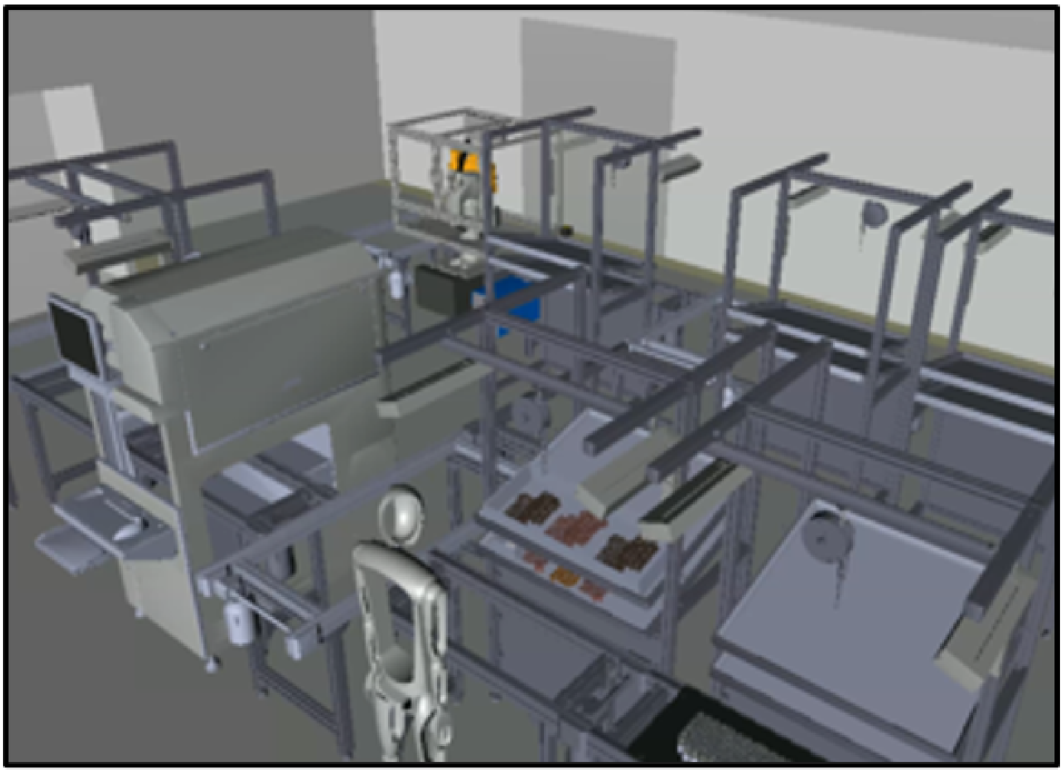


Figure 4.2: Pilot battery module assembly facility modelled in vueOne.

Table 4.1: Allocation of operations to workstations.

Station number	Operation number	Operation name
Station 1	Operation 1	Cell testing
	Operation 2	Cell loading
Station 2	Operation 3	Thermistor assembly
	Operation 4	Cooling system assembly
Station 3	Operation 5	Plastic welding
	Operation 6	Busbar assembly
Station 4	Operation 7	Pulse arc welding
Station 5	Operation 8	Ultrasonic wire bonding

follows: testing (operation one) and cell loading (operation two) are performed in workstation one, thermistor assembly (operation three) and cooling system assembly (operation four) are performed in workstation two, plastic welding (operation five) and busbar assembly (operation six) are performed in workstation three, pulse arc welding (operation seven) for variant A is performed in workstation four and ultrasonic wire bonding (operation eight) for variant B is performed in workstation five. The operation allocation is also presented in **Table 4.1**. Each of the five workstations perform different assembly operations and hence it can be considered that there are five workstation types.

According to the process allocation, station types four and five perform the battery welding process, however, they differ in the technology of welding. Therefore, workstation type four is bypassed by module B which does not require pulse-arc welding. Similarly, workstation type five is bypassed by module A since it does not require wire bonding. The length and width is one meter for both workstations and all stations have the same footprint and are modular. Product transfer on assembly completion is achieved using conveyor belts that are used to transfer the products from one station to another using pallets; buffers to store products between stations are not available. The pallets have RFID tags to track the status of the product as it moves through the line. If a product is found to be faulty, the status will be marked as *'rework'* and it will, therefore, bypass all other stations and will subsequently be removed from the pilot line.

The battery cells are received in cartons from the supplier and the carton for product variant A has five rows of 24 cells each and the carton for product variant B has five rows of 18 cells each. Product variant A comprises of cylindrical cells of *'18650'* type and variant B comprises of cylindrical cells of *'21700'* type. Due to confidentiality of the product designs, a detailed explanation of the product is not provided but a generic description of the Li-ion battery module design is discussed. A battery module comprises of the battery, busbar, electrical connections and insulations, cooling system and the battery module frame [KUKA, 2020]. The battery geometry can be cylindrical, prismatic or pouch and the electrical connections are achieved by welding busbars onto the batteries. In case of cylindrical batteries, the numbers *'18650'* and *'21700'* are used to represent the battery diameter and height. For example, 18650 represents a cylindrical Li-ion battery having a diameter of 18mm and height of 65mm. The electrical connections are achieved using welding processes such as pulse arc, ultrasonic and laser welding and the cooling system could be liquid, solid or air-based cooling [Saariluoma et al., 2020]. All these components play an important role in the design of the battery module. With this, the explanation of the assembly facility is complete and hence, the next paragraph explains the two processes that are considered for demonstration of the DDSM methodology. The task sequences of operation one and operation two are represented in **Figures 4.3** and **4.4**, respectively. When applying DDSM to the case study, the first step is to model and encapsulate the data pertaining to the five workstation types in the kinematic modelling software. To scale-up the pilot line, it is necessary to have a new target daily demand; this is 65 products of variant A and 65 products of variant B and the current daily production volume is 20 products of variant A and 20 products of variant B.

Task Sequence (Product A)	**Task Sequence (Product B)**
1.1 Move to position (***x1,y1,z1***)	1.1 Move to position (***x6,y6,z6***)
1.2 Test battery cell (*30)	1.2 Test battery cell (*30)
1.3 Repeat steps 1.1 and 1.2 – (offset 120mm in x) (*4)	1.3 Repeat steps 1.1 and 1.2 – (offset 120mm in x) (*3)

Figure 4.3: Task sequence for operation one (cell testing).

This concludes the basic overview of the case study. The operation sequence, product variants, demand, assembly facility and the application were explained in detail. The next section explains the application of the DDSM methodology to the battery module assembly case study.

4.2 Demonstration of methodology: Stage one

The stage one of the methodology comprises of kinematic modelling module, knowledge representation module and workstation configuration selection. For the purpose of demonstrating the DDSM methodology, workstation one is selected and the workflow is explained in detail. Although other workstations are also modelled in vueOne for process time calculation, they are not explored beyond the kinematic modelling stage. In other words, the alternate workstation configurations for workstations other than station one are not considered.

Task Sequence (Product A)	**Task Sequence (Product B)**
2.1 Move to position (***x11,y11,z11***)	2.1 Move to position (***x21,y21,z21***)
2.2 Hold battery cell	2.2 Hold battery cell
2.3 Move to position (***x12, y12,z12***)	2.3 Move to position (***x22, y22,z22***)
2.4 Release battery cell	2.4 Release battery cell
2.5 Repeat steps 2.1 to 2.4 – (offset 20mm) (*24)	2.5 Repeat steps 2.1 to 2.4 – (offset 20mm in x) (*18)
2.6 Move to position (***x13,y13,z13***)	2.6 Move to position (***x23,y23,z23***)
2.7 Hold battery cell	2.7 Hold battery cell
2.8 Move to position (***x14, y14,z14***)	2.8 Move to position (***x24, y24,z24***)
2.9 Release battery cell	2.9 Release battery cell
2.10 Repeat steps 2.6 to 2.9 – (offset 20mm in x) (*24)	2.10 Repeat steps 2.6 to 2.9 – (offset 20mm in x) (*18)
2.11 Move to position (***x15,y15,z15***)	2.11 Move to position (***x25,y25,z25***)
2.12 Hold battery cell	2.12 Hold battery cell
2.13 Move to position (***x16, y16,z16***)	2.13 Move to position (***x26, y26,z26***)
2.14 Release battery cell	2.14 Release battery cell
2.15 Repeat steps 2.11 to 2.14 – (offset 20mm in x) (*24)	2.15 Repeat steps 2.11 to 2.14 – (offset 20mm in x) (*18)
2.16 Move to position (***x17,y17,z17***)	2.16 Move to position (***x27,y27,z27***)
2.17 Hold battery cell	2.17 Hold battery cell
2.18 Move to position (***x18, y18,z18***)	2.18 Move to position (***x28, y28,z28***)
2.19 Release battery cell	2.19 Release battery cell
2.20 Repeat steps 2.16 to 2.19 (offset 20mm in x) (*24)	2.20 Repeat steps 2.16 to 2.19 (offset 20mm in x) (*18)
2.21 Move to position (***x19,y19,z19***)	2.21 Move to position (***x29,y29,z29***)
2.22 Hold battery cell	2.22 Hold battery cell
2.23 Move to position (***x20, y20,z20***)	2.23 Move to position (***x30, y30,z30***)
2.24 Release battery cell	2.24 Release battery cell
2.25 Repeat steps 2.21 to 2.24 – (offset 20mm in x) (*24)	2.25 Repeat steps 2.21 to 2.24 – (offset 20mm in x) (*18)

Figure 4.4: Task sequence for operation two (cell loading).

4.2.1 Modelling in vueOne

For the virtual modelling, a combination of control and non-control components are employed in the model building process. Specifically, for workstation one which has testing and cell loading operations, the primary control components utilised include the actuators, human resource, robot and grippers. The non-control components include the station frame, fixtures, pallet stoppers, pallets and shelves.

Operation one

From **Figure 4.3**, the operation one task sequence for product A comprises of two types of tasks that are a combination of testing battery cells and moving them. Similarly, product B also has the same tasks but the difference is the module design and the number of cells being assembled. In order to model the operations in vueOne, it is important to understand the various equipment that are required for the operation. Operation one comprises of testing battery cells and for this purpose, a Hioki multiplex cell testing machine is used in the pilot line. The vueOne model just replicates the tasks performed in the pilot line using the cell testing equipment CAD and actuators to lift and lower the testing system to the cell cartons. Translation kinematics are defined on the actuators as they move along the *'z axis'*.

Operation two

From **Figure 4.4**, there are 25 tasks that need to be modelled. **Figure 4.5** represents the kinematic model of operation two alongside the real system. Operation two utilises a robot in the pilot line and hence, it is necessary to do the same in the model. Therefore, the *'V-Rob'* module is used for this model. The V-Rob module has a pre-defined library of robots and a general purpose ABB robot is selected. This saves time since the robot does not have to be modelled from scratch. The grippers, in the real system, are 3-finger grippers that are capable of holding cylindrical cells. The gripper CAD is imported into the model and the translation kinematic are defined on all three fingers such that they operate simultaneously when the signal is received. The robot picks the battery cells from the cell carriers that are available on either sides and loads them into the battery module.

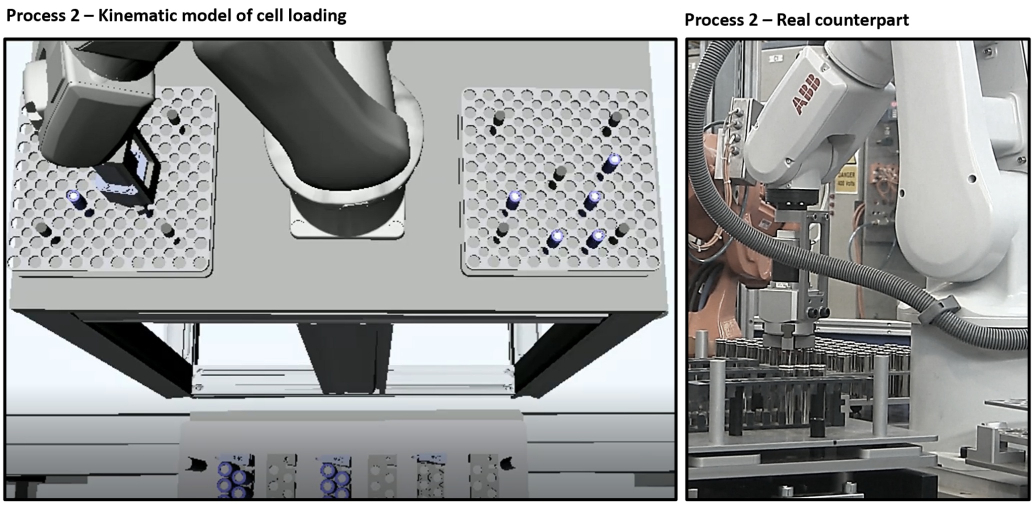


Figure 4.5: Kinematic model of operation two (cell loading) in vueOne.

Validating the model

After the virtual model is completed, it was validated using the time taken for the cell loading operation. The cell loading operation is performed in workstation one and it comprises of the tasks and sequence mentioned in **Figure 4.4**; considering product variant A, it took approximately 10 minutes for this assembly operation in the kinematic model. Comparing this with the actual process time, there was a difference of about 30 seconds. This difference might be due to the component weight, friction, offset in position, etc., that are not considered in the virtual model. The deviation is still considered acceptable for the considered proof of concept to demonstrate the methodology.

Model output data

Following the virtual model creation, the next step is to export the data which is part of dataset (a) to the common database. This includes data such as the workpiece attributes, operation sequence, number of operations, task types, station footprint, axis of motion and process parameters. This dataset is categorised into three different XML representations for the workpiece, process and resource data, respectively. They are represented more clearly in **Figures 4.6** and **4.7**. The next paragraph demonstrates the use of the kinematic modelling module for selecting equipment for the case study.

4.2.2 Knowledge representation module

The above-mentioned workpiece, process and resource data will be used for the equipment selection within the knowledge representation module. Especially, the process data such as process parameters, sequence and task types are utilised to perform the query using the Protégé workflow chart that was introduced in section 3.3.2. Using the flow chart, the eight operations that are distributed across the workstations are considered one by one.

Workpiece XML


```xml
<Workpieces>
  <Workpiece>
    <WorkpieceID>WP343VDV</WorkpieceID>
    <WorkpieceName>18650</WorkpieceName>
    <DimensionUnit>mm</DimensionUnit>
    <WeightUnit>kg</WeightUnit>
    <Description>Cylindrical cell</Description>
    <Length>NA</Length>
    <Width>NA</Width>
    <Height>65</Height>
    <Diameter>18</Diameter>
    <Weight>0.048</Weight>
    <Material>Metal</Material>
  </Workpiece>
  <Workpiece>
    <WorkpieceName>21700</WorkpieceName>
    <WorkpieceID>WP867VDS</WorkpieceID>
    <Description>Cylindrical cell</Description>
    <DimensionUnit>mm</DimensionUnit>
    <WeightUnit>kg</WeightUnit>
    <Length>NA</Length>
    <Width>NA</Width>
    <Height>70</Height>
    <Diameter>21</Diameter>
    <Weight>0.07</Weight>
    <Material>Metal</Material>
  </Workpiece>
</Workpieces>
```


Process XML


```xml
<Process>
  <Operation>
    <OperationName>Operation1</OperationName>
    <OperationID>P6465IGU</OperationID>
    <Description>Testing of battery cells</Description>
    <SequenceNumber>1</SequenceNumber>
    <AssemblyDirection1>X</AssemblyDirection1>
    <AssemblyDirection2>Y</AssemblyDirection2>
    <AssemblyDirection3>Z</AssemblyDirection3>
    <TaskType>Move,Test</TaskType>
    <RangeInX>100</RangeInX>
    <RangeInY>100</RangeInY>
    <RangeInZ>50</RangeInZ>
    <DimensionUnit>mm</DimensionUnit>
    <Payload>NA</Payload>
    <WeightUnit>kg</WeightUnit>
    <PartGripMinDia>NA</PartGripMinDia>
    <PartGripMaxDia>NA</PartGripMaxDia>
  </Operation>
  <Operation>
    <OperationName>Operation2</OperationName>
    <OperationID>P2458HJ</OperationID>
    <Description>Loading battery cells</Description>
    <SequenceNumber>2</SequenceNumber>
    <AssemblyDirection1>X</AssemblyDirection1>
    <AssemblyDirection2>Y</AssemblyDirection2>
    <AssemblyDirection3>Z</AssemblyDirection3>
    <TaskType>Move,Hold,Release</TaskType>
    <RangeInX>100</RangeInX>
    <RangeInY>100</RangeInY>
```


Figure 4.6: XML representation of the workpiece and process data from kinematic model.

Resource XML


```xml
<Resources>
  <Resource>
    <ResourceName>TestingStation</ResourceName>
    <ResourceID>RE543VDG</ResourceID>
    <Description>Testing of battery cells</Description>
    <Footprint>1000*1000*800</Footprint>
    <DimensionUnit>mm</DimensionUnit>
    <AllowableWeight>200</AllowableWeight>
    <WeightUnit>kg</WeightUnit>
  </Resource>
  <Resource>
    <ResourceName>CellLoadingStation</ResourceName>
    <ResourceID>RE754DGFS</ResourceID>
    <Description>Loading of battery cells</Description>
    <Footprint>1000*1000*800</Footprint>
    <DimensionUnit>mm</DimensionUnit>
    <AllowableWeight>200</AllowableWeight>
    <WeightUnit>kg</WeightUnit>
  </Resource>
</Resources>
```


Figure 4.7: XML representation of the resource data from kinematic model.

Protégé: Operation one (cell testing)

The activities in the ontology editor start with operation one, *'O'* = *1*, which comprises of the battery testing. From **Figure 4.8**, it can be seen that operation one comprises of the test task which is not within the defined five tasks. The ontology, in its current form, does not support the selection of testing equipment since there are more parameters and considerations that need to be included in the knowledge representation for the testing tasks. Therefore, the existing testing time of 30

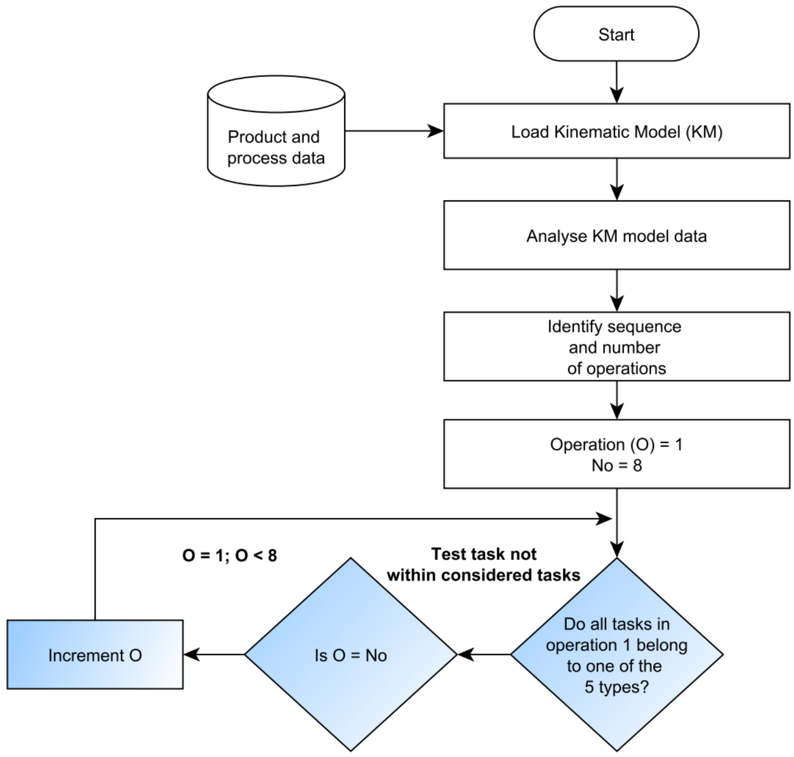


Figure 4.8: Protégé flow chart for operation one.

seconds obtained from the pilot line will be added to the process time and this is explained in section 4.2.2 under validating the query results. Accordingly, operation one will not be considered for further analysis and the value of *‘O’* is increased to 2. This process is now repeated for operation two.

Protégé: Operation two (cell loading)

Advancing to operation two, the cell loading process has move and hold/release tasks. Since they are within the defined five tasks, operation two is considered for further analysis and **Figure 4.9** represents this. As explained previously, operation one and two are only considered in detail for demonstration of the methodology. As operation two is eligible for further analysis, it is necessary to perform the query process. For this purpose, the information such as product weight, dimensions, assembly directions, batch size, gripping force required, drive type, gripping distance, repeatability, accuracy, gripper range, payload, space available in the workstation, and allowable weight are obtained from the database.

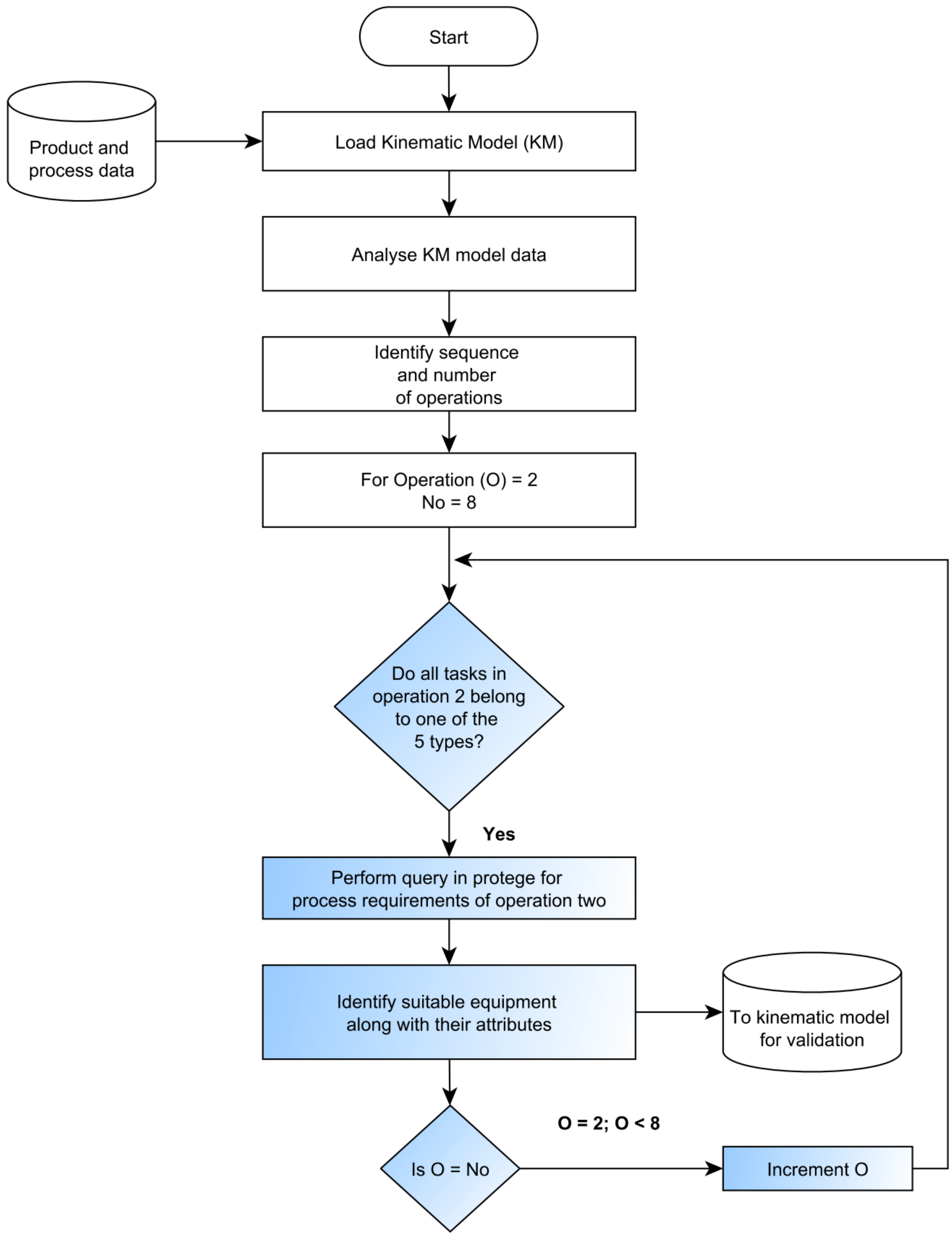


Figure 4.9: Protégé flow chart for operation two.

Operation two: Query

The query is designed considering the process parameters for both product variant A and B. The query design depends on the number of different task types in an operation. If there is only one task type, the query comprises of just finding the equipment that performs that particular task. However, when two or more task types exist in the same operation, the query comprises of more sections. Consider an example of move and hold tasks that are part of an operation; it is necessary to find an equipment or a combination of equipment that performs the tasks. The query comprises of three sections: finding an equipment that performs the move task, finding an equipment that performs the hold task, and combining them into a single set of results. It is important to note that the number of sections in the query design might increase with the number of task types in the process.

The following process parameters were considered for the query design. The weight of the battery cell ranges from 45g for 18650 cells and 70g for 21700 cells. Therefore, an actuator or robot that can handle these cells should be selected. Additionally, a gripper which is capable of picking up the cells according to their dimensions needs to be selected. Moreover, there are three axes of motion, x,y and z which need to be considered for the assembly during the pick and place process. For selecting equipment that perform the move task, there are different parameters such as the axis of motion, the range of motion, repeatability, accuracy, etc., that need to be considered.

Considering the gripping of the cells, different parameters such as gripper stroke, gripper payload, gripper weight and gripping force can be considered. It can be seen that these parameters are not necessarily associated with the equipment that perform the move task. Therefore it is necessary to separate the equipment for the two different task types; hence the need to split the query into different sections. The first and second sections are for finding components that perform the move tasks and hold/release tasks, respectively. Section three is for combining the results of the first and second sections. In order to initially create the two separate sets of results, the *'sqwrl:makeSet'* is used; one set for each task type is created and the *'sqwrl:union'* is used to combine both sets together at the end of the query.

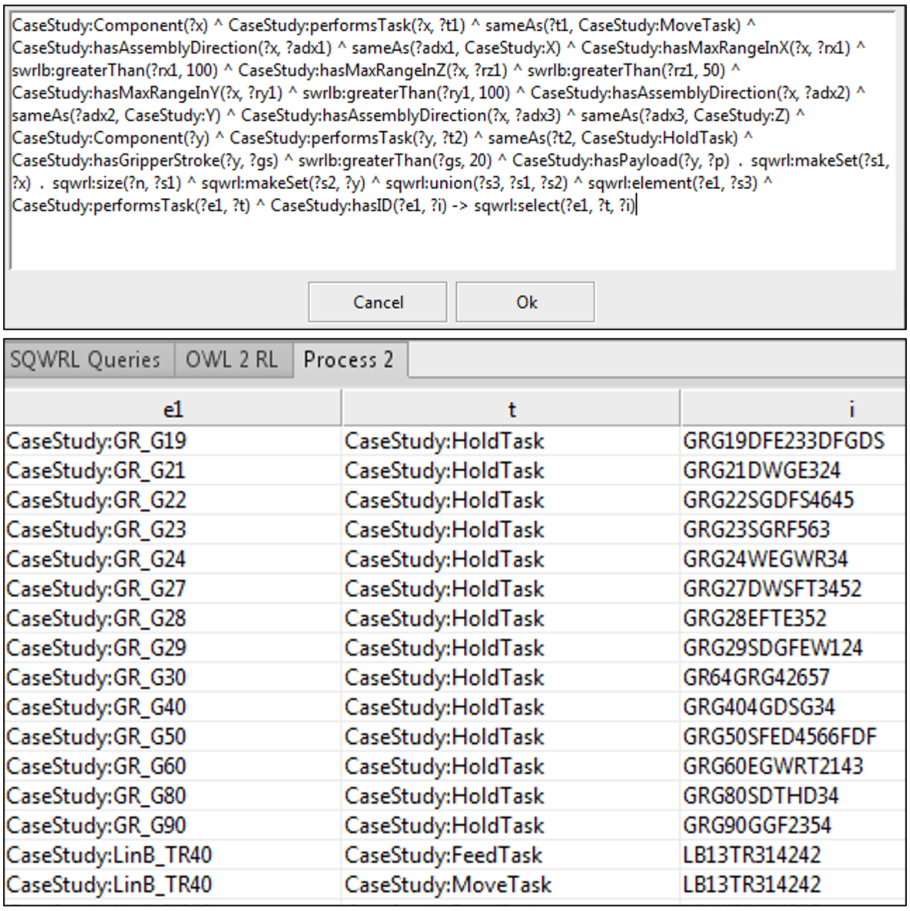


Figure 4.10: The SQWRL query and results in Protégé.

The degree of freedom and working range are considered as primary criteria for selecting the equipment that perform the move task; the selected equipment form part of set one. The payload and gripper stroke are considered as primary criteria for selecting equipment that perform the hold task; the selected equipment form part of set two. The components from both sets are combined using the *'sqwrl:union'* function to provide a total of 30 components that meet the defined criteria. **Figure 4.10** shows a few of the components that meet the criteria. An XML representation of the results is provided in **Figure 4.11**.

Query results


```xml
<QueryResults>
  <Process>
     <OperationName>Operation2</OperationName>
     <OperationID>P2458HJ</OperationID>
     <Description>Loading battery cells</Description>
     <Configuration>1</Configuration>
     <ConfigurationID>WS1GRG24LB13</ConfigurationID>
     <TimeUnits>seconds</TimeUnits>
     <WeightUnit>kg</WeightUnit>
     <Equipment1>Gantry</Equipment1>
     <EquipmentID>LB13TR314242</EquipmentID>
     <CapitalCost>4000</CapitalCost>
     <Task>Move</Task>
     <Brand>GH</Brand>
     <Payload>4</Payload>
     <Weight>100</Weight>
     <Equipment2>Gripper</Equipment2>
     <EquipmentID>GRG24WEGWR34</EquipmentID>
     <Task>Hold/Release</Task>
     <Payload>1</Payload>
     <Brand>YO</Brand>
     <CapitalCost>5000</CapitalCost>
     <Weight>0.800</Weight>
   </Process>
  <Process>
     <OperationName>Operation2</OperationName>
     <OperationID>P2458HJ</OperationID>
     <Description>Loading battery cells</Description>
     <Configuration>2</Configuration>
     <ConfigurationID>WS1GRG28LB15</ConfigurationID>
```


Figure 4.11: XML representation of the results of the query.

Validating the query results

The query results need to be validated in the vueOne toolset by creation creation of kinematic models and checking for collision and process time. For validation, from the set one, *'three axis gantry and delta robot'* are selected. From set two, two different *'vacuum grippers'* are considered for modelling in *'vueOne'*. This process of selecting specific equipment for further validation is according to user preferences. Workstation configuration 1, referenced as *'WS1GRG24LB13'*, is obtained by pairing the vacuum gripper and gantry. The IDs of the vacuum gripper and gantry are *'GRG24WEGWR34'* and *'LB13TR314242'*, respectively. Similarly, the second vacuum gripper and delta robot having IDs *'DB434DGSH'* and *'GR4668HTDSD3'*, respectively, form workstation configuration two with an ID *'WS1GRG28LB15'*. Both workstation configurations are associated with their respective workstation KPIs using the references IDs. The two workstation configurations are validated and their images are shown in **Figure 4.12**

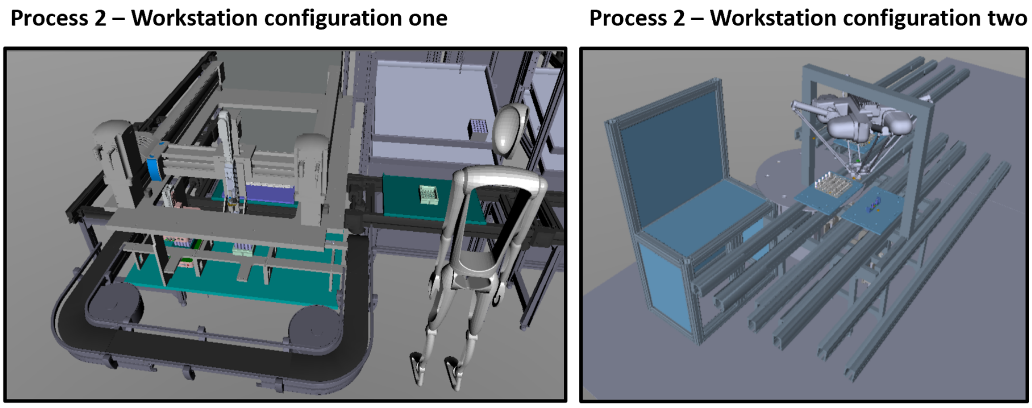


Figure 4.12: Validation of the query results.

Workstation configuration results


```xml
<WorkstationConfigurations>
  <WorkstationConfiguration>
    <OperationName>Operation2</OperationName>
    <OperationID>P2458HJ</OperationID>
    <Description>Battery cell loading</Description>
    <ConfigurationNumber>1</ConfigurationNumber>
    <ConfigurationID>WS1GRG24LB13</ConfigurationID>
    <EquipmentID>LB13TR314242_GRG24WEGWR34</EquipmentID>
    <Time_A>360</Time_A>
    <Time_B>270</Time_B>
    <TimeUnit>seconds</TimeUnit>
    <CapitalCost>9000</CapitalCost>
    <MTBF>5000</MTBF>
    <MTTR>300</MTTR>
    <OperatorRequired>1</OperatorRequired>
  </WorkstationConfiguration>
  <WorkstationConfiguration>
    <OperationName>Operation2</OperationName>
    <OperationID>P2458HJ</OperationID>
    <Description>Battery cell loading</Description>
    <ConfigurationNumber>2</ConfigurationNumber>
    <ConfigurationID>WS1GRG28LB15</ConfigurationID>
    <EquipmentID>DB434DGSH_GR4668HTDSD3</EquipmentID>
    <Time_A>120</Time_A>
    <Time_B>90</Time_B>
    <TimeUnit>seconds</TimeUnit>
    <CapitalCost>20000</CapitalCost>
    <MTBF>5000</MTBF>
    <MTTR>300</MTTR>
    <OperatorRequired>1</OperatorRequired>
```


Figure 4.13: XML representation of the workstation configuration results.

Through the validation of selected configurations in vueOne, the new configurations are visualised and the workstation process time is calculated. The process time calculation is done with the motion time data available in data sheets and the inherent capability of kinematic modelling tools to calculate the process time using the kinematic behaviour of the equipment. It was found that the time taken to per-

Workpiece data in database

WorkpieceID	WorkpieceName	DimensionUnit	WeightUnit	Description	Length	Width	Height	Diameter
WP343VDV	18650	mm	kg	Cylindrical cell	NA	NA	65	18
WP867VDS	21700	mm	kg	Cylindrical cell	NA	NA	70	21

Process data in database

OperationNa	OperationID	Description	SequenceNumber	AssemblyDire	AssemblyDire	AssemblyDire	TaskType	RangeInX	RangeInY
Operation1	P6465IGU	Testing of batte	1	X	Y	Z	Move,Test	100	100
Operation2	P2458HJ	Loading battery	2	X	Y	Z	Move,Hold,Release	100	100

Resource data in database

ResourceName	ResourceID	Description	Footprint	DimensionUnit	AllowableWeig	WeightUnit
TestingStation	RE543VDG	Testing of battery cells	1000*1000*800	mm	200	kg
CellLoadingStation	RE754DGFS	Loading of battery cells	1000*1000*800	mm	200	kg

Workstation design table

Operation Number	Operation ID	Configuration Number	Configuration ID	Equipment ID	Time (product A)	Time (product B)	Units	Capital Cost	MTBF	MTTR	Operator Requirement
2	*P2458HJ*	1	*WS1GRG24LB1*	*LB13TR314242_ GRG24WEGWR34*	360	270	*seconds*	9000	5000	300	1
2	*P2458HJ*	2	*WS1GRG28LB1*	*DB434DGSH_ GR4668HTDSD3*	120	90	*seconds*	20000	5000	300	1

Figure 4.14: Workstation design table and workstation configuration data in the common database.

form operation two in workstation configuration one is 360 seconds for product A and 270 seconds for product B. Similarly, the time taken to perform operation two in configuration two is 120 seconds for product A and 90 seconds for product B. Since for operation one, which is the cell testing operation, the Protègè workflow did not identify alternate equipment, the current pilot line testing time of 30 seconds is added to the time taken for operation two, to get the total process time for workstation one.

In addition to the process time calculation, the possibility of the gantry or robot colliding with other components of the workstation is also assessed. An XML representation of the workstation configuration results is presented in **Figure 4.13**. The workstation design table is shown in **Figure 4.14**, following which the data such as process time, cost, MTBF, MTTR, etc., are stored in the common database.

4.3 Demonstration of methodology: Stage two

The Stage two of DDSM methodology involves the system configuration selection process. As a first step, the battery module assembly line is virtually modelled in DES software such that it represents the existing pilot line. **Figure 4.15** represents the DES model of the pilot line under study.

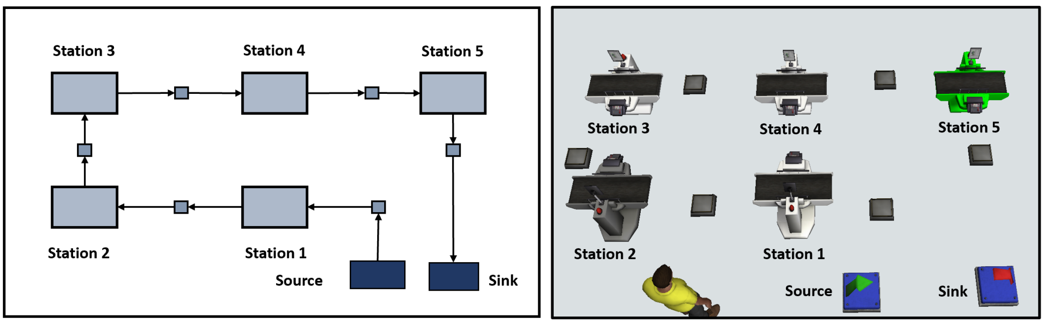


Figure 4.15: Pilot line model for battery module assembly case study in FlexSim.

4.3.1 DES model module

Model building

The assembly system that is considered, is a mixed model assembly line and all workstations are assumed to be available at time zero of the simulation model. The workstations are arranged in a sequential manner with the AGVs transporting products between the stations. In the existing pilot line, conveyors transfer parts between the stations. However, to support the scale-up modelling, the conveyors are replaced with AGVs such that there is more flexibility in the model. This is because the AGV can be programmed to cater to whichever workstations are present within its network and it does not need to follow a defined path. This is beneficial for simulation optimisation since the addition or removal of workstations from the DES models automatically alters the AGV allocation and routing to workstations; this is not the case when conveyors are used in the simulation model. Buffers are allocated between the stations for part storage and can hold a maximum of five products. The stations can be bypassed if a product variant does not need to be processed in it. First-in-first-out (FIFO) scheduling policy is considered for transfer of products from buffers to workstations.

Five workstation types, corresponding to the allocation in **Table 4.1** are modelled in DES and each workstation assembles only one product at a time. Similarly, AGVs can transport only one part at a time. The source represents the location where the battery cartons enter the model and the sink represents the location where the modules that have been assembled are stored. Each workstation has setup time for product changeover which is assumed to be the same for changing from product A to product B and vice-versa. Preemption of operators who are already working on a specific job is not allowed and once the operators start working on a product, they remain in the corresponding workstation until the operations on that product are finished. The AGVs that are used for transportation have control points where they are charged; they return to these points on completion of transportation tasks. Both AGVs and operators are monitored using the task executor which allocates the job on a FIFO basis. Therefore, both AGV and operator are free to work on any workstation and are not restricted to any particular region of the production system.

Operators are assumed to be multi-skilled; operators and AGVs are assumed to be always available excluding the break times. Stochasticity is introduced in the model using probability distributions. Five different aspects of the system where probability is introduced are i) part arrival, ii) process and setup times, iii) downtime, iv) time between failure, and v) first time failure. The process time, setup times and down times follow triangular distribution [Banks, 1998], whereas part arrival follows exponential distribution. These distributions are obtained based on literature and data from pilot line. After several experimental runs, a warm-up time of 2500 seconds was found suitable for the model; the warm-up time enables the calculation of statistics in the simulation model once it reaches steady state [Robinson, 2007]. The total simulation time that represents a single shift is 28800 seconds and only one shift is modelled. The sub-components and raw material required for the assembly are assumed to be always available.

Verification and validation

After the model creation, it is verified, tested and validated. The model verification generally involves checking whether the model is built correctly. Model validation involves ensuring that the model behaviour is consistent with the real production system behaviour. This section is primarily referenced from [Banks, 1998]. There are over 75 techniques for verification, validation and testing that are classified into informal, static, dynamic and formal. However, a number of these techniques are data-intensive. Considering the fact that this simulation study is done in the early design phases, subjective approaches for verification and validation are employed. Firstly, the informal technique of *'desk checking'* is employed. Desk checking involves either the self-examination of the work or peer review to ensure correctness, completeness, consistency along with checking the syntax and code of any logic or algorithm that was built within the DES model. Additionally, since FlexSim enables the visualisation of the simulation run, visual verification of the model was also was done. After ensuring that the model is correct, face validation was done with members of the project; it is an informal validation technique where the people knowledgeable in the subject area use their experience and intuition to assess the model and its results [Banks, 1998]. The time values from the pilot line were used in the simulation run and the validation parameter was selected as the number of battery modules produced daily when four operators are employed. The results were found consistent with the real pilot line scenarios and this was also reviewed with two field experts and three project members.

Parametric DES model

Once the skeleton model is created, it is important to ensure that it is parametric. The DES model will be updated with *i*) information from Stage one pertaining to the workstation processing and setup time, cost of workstations and IDs of the selected candidates, and *ii*) the values of the decision variables from MATLAB to generate scale-up solutions. The various parameters that can be tuned from outside FlexSim are as follows: the workstation quantity, operator quantity, AGV quantity, station process time and station setup time.

The available space in the layout is divided into *'slots'*. The workstations are allocated to *'slots'* and when new workstations of the same type need to be added to the model, they are assumed to be added in parallel to the existing ones. In other words, each of the five workstation types can have copies of the same to improve productivity, which is represented using five decision variables. Each variable corresponds to one workstation type and it refers to the quantity of the respective workstation type. For instance, if the second decision variable from MATLAB has a value of two, it means that workstation type two has another replica in parallel that performs the thermistor assembly (operation three) and cooling system assembly (operation four). The decision variables are explained in more detail in the next section.

Connecting FlexSim and KEPServerEX

This section explains the details of the KEPServerEX and FlexSim connection. The local server is first created and the device and groups are defined within it. Within the group, various *'tags'* can be added and their name plays an important role in establishing the link with both MATLAB and FlexSim. The *'emulator'* tool in FlexSim allows creation of two types of variables, those that need to be read and those that need to be written. The decision variable values from the MATLAB need to be read by FlexSim. On the other hand, the KPIs such as system throughput that are necessary for objective function evaluation need to be written by FlexSim.

Since the DES model is parametric, all time-related, maintenance-related values and other resource-related parameters can be stored in the form of a *'Global Table'* in FlexSim. In this way, all necessary input sources for FlexSim are established. The run speed and model termination time can be controlled from outside FlexSim using a batch file. The use of the termination time enables automation of the simulation optimisation process since the MATLAB optimisation can be continued only when the execution of FlexSim is stopped. For each optimisation iteration in MATLAB, ten replication are of the experiment are done within FlexSim. The average of the throughput values across these ten replications for products A and B is passed back to MATLAB. These replications are very important as they impact the convergence of the simulation optimisation; based on trial and error, it was found that ten replications were sufficient for good convergence to a pareto front for the considered case study.

4.3.2 Optimisation module

Following the creation of the parametric DES model, the optimisation problem is formulated in MATLAB and for this purpose, several *'functions'* need to be written.

Step 1: Fitness function

Starting with the core *'optimisation algorithm'*, the first step is to create the fitness function which evaluates the score of a particular population with respect to the objective function. As explained in Chapter three, two conflicting objectives, scale-up cost and throughput are considered. For objective one, the aim is to minimise the scale-up cost and for objective two, the aim is to maximise the throughput. However, since MATLAB aims to minimise the objective functions, objective two is modified to minimise (1/throughput).

Step 2: Decision variables

The next step is to define the number and parameter of the decision variables. The four types of decision variables considered for the optimisation study are x_i^1, x_j^2, x_k^3, x_l^4 that represent the number of each type of workstation, number of each type of MHU, number of each type of operator, and the number of workstations that have alternate configurations, respectively. For this case study, because there are five different types of workstations, in x_i^1, the value of i ranges from one to five. Considering the variable x_j^2, only one type of MHU is considered and hence the value of j is one. For the variable x_k^3, only one type of operator is considered and hence the value of k is one. For the last variable type, the workstation configuration selection was done in the ontology editor only for the first workstation and hence only workstation one has alternate configurations; therefore, the value of the variable l is one.

In total, there are eight decision variables and the memory load that is brought about due to the simulation optimisation restricts the total number of decision variables that can be considered. All eight decision variables considered are integers and hence a multi-objective simulation optimisation with integer GA is selected.

Step 3: Boundary conditions

Following this, the upper bound and lower bound for the decision variables are set as shown in **Table 4.2**. A total of 22 slots are considered for the workstations and this restricts the maximum number of workstations that can be accommodated. If the variables x_1^1 to x_5^1 have the upper bound values of five, then the total number of workstations exceeds the available space. To overcome this, it is possible to add inequality constraints in the algorithm. However, it is not advisable to add the inequality constraint whilst already having integer constraints in the MATLAB GA algorithm. Hence, for those iterations where the number of workstations exceed the available space, a penalty cost is added to the scale-up cost. In this way, such iterations will not be considered as good solutions and will be removed from the solution space.

Step 4: Optimisation options

As a next step, the various genetic algorithm solver options such as the initial population creation, mutation function, crossover function, maximum stall generations, maximum generations, *'pareto fraction'* and plot functions are considered. For the initial population creation, mutation, crossover functions, instead of writing the

Table 4.2: Decision variables and their values.

Variable	Description	Lower bound	Upper bound
x_1^1	number of workstations of type 1	1	5
x_2^1	number of workstations of type 2	1	5
x_3^1	number of workstations of type 3	1	5
x_4^1	number of workstations of type 4	1	5
x_5^1	number of workstations of type 5	1	5
x_1^2	number of MHUs of type 1	1	2
x_1^3	number of operators of type 1	1	6
x_1^4	workstation configuration for workstation type 1	1	2

code from scratch, the default functions provided by MATLAB for integer GA is used. A *'population size'* of 20 and a *'maximum generation limit'* of 100 is selected. Ten simulation repetitions for each evaluation or iteration are considered. The *'stall generation limit'* is set to 15. The settings are decided after experimentation and are found sufficient to provide the required set of non-dominated solutions. The pareto fraction is set as 0.7 and the default settings used for distance calculation and function tolerance for pareto spread are *'phenotype'* and 1e-4, respectively.

To monitor the the progress towards convergence, the *'best fitness scores'* for both objective functions are plotted at the end of each generation. The diversity of the pareto front is checked by the measuring the distance and pareto spread. The distance measurement ensures even spread of solutions on the pareto front, provided it is continuous. The average change in the pareto spread over the *'MaxStallGenerations'* is a parameter that terminates the optimisation on satisfying the stopping criteria. If this average change is less than the function tolerance value of 1e-4, then optimisation will be terminated. For a diverse pareto front, it is expected that the average distance measure and spread of pareto front have low values.

Function: FlexSim initialisation

Now that the core optimisation algorithm is defined, two other functions are written in MATLAB to support the core function. Both these functions are run for every single iteration and they need to be provided with the new values of the decision variables for that particular iteration. The first function is the FlexSim initialisation function that is used to initialise the *'batch file'* that starts the simulation. The pseudocode for the FlexSim initialisation function is shown in **Table 4.3**. It starts with the creation of *'daobj'* to connect to the server. This is followed by the creation of a *'Group'* to store the decision variables. Step three, from **Table 4.3**, is very important for establishing the link between the decision variables in MATLAB to the *'tags'* in KEPServerEX. In this step, the decision variables are defined. The *'Device'* and *'Group'* mentioned in step three represent the elements in the KEPServerEX and the *'AGVQty'* represents the tag in the server. The next step is to store the values of the decision variables that are decided by MATLAB for each iteration in the *'AGVQty'* item object. The *'flexin'* vector represents the values of the decision variables decided within MATLAB. The last step is to write a code to start FlexSim from MATLAB, for which the batch file is used.

As shown in **Figure 4.16**, in MATLAB OPC toolbox, one object is created for each variable that needs to be communicated to Flexsim. For instance, the variable that represents the number of MHUs is x_j^2 and only AGV is considered for material handling and transportation then j is equal to one. This variable can take up different values for each iteration of the optimisation and this needs to be communicated to Flexsim. Therefore, this particular variable i.e. x_1^2 is created as an *'item object'* in the MATLAB OPC toolbox (Steps 1, 2 and 3). After creation of this object, it then needs to be written to KEPServerEX using the write function (Step 4). In this way, all eight decision variables are linked to KEPServerEX. Certain other variables such as process time, maintenance and cost can be added to a lookup table within Flexsim using the *'Global Table'* tool.

Table 4.3: FlexSim initialisation function.

Initialisation function	
(1)	Create ‘daobj’ to connect to server using OPC UA protocols; **daobj = opcda(‘localhost’,‘Kepware.KEPServerEX.V6’);**
(2)	Create group for item objects; this represents the decision variables; **Grp = addgroup(daobj,‘Group’)** **set(Grp,‘LogFileNAme’,‘opcdatalog.olf’);**
(3)	Create item objects for eight variables within created group and set their datatype; **AGVQty = additem(Grp,‘MFConnection.Device.Group.AGVQty’);** **set(AGVQty,‘DataType’,‘int16’);**
(4)	Write the values for the decision variables; **write(AGVQty,flexin(6));**
(5)	Run batch file to start Flexsim; **command = “C:\Users\RunFlexsim.bat”;** **[status,cmdout] = system(command);**

Function: Fitness evaluation

The fitness evaluation function is essential for calculating the scores of the objective function. From Chapter three, the objective functions are represented in two equations, 3.1 and 3.2. The equations are utilised in this function to calculate what is known as the fitness score. This is essentially the value obtained while solving the two equations. The steps involved in the function are explained as follows:

1. The *‘flexin’* vector that has the values of the eight decision variables is the input for this function. The equation 3.1 which represents the scale-up cost is considered.

2. It comprises of four elements of cost: the processor cost, material handling cost, operator-related cost and penalty cost. The values of the first three cost

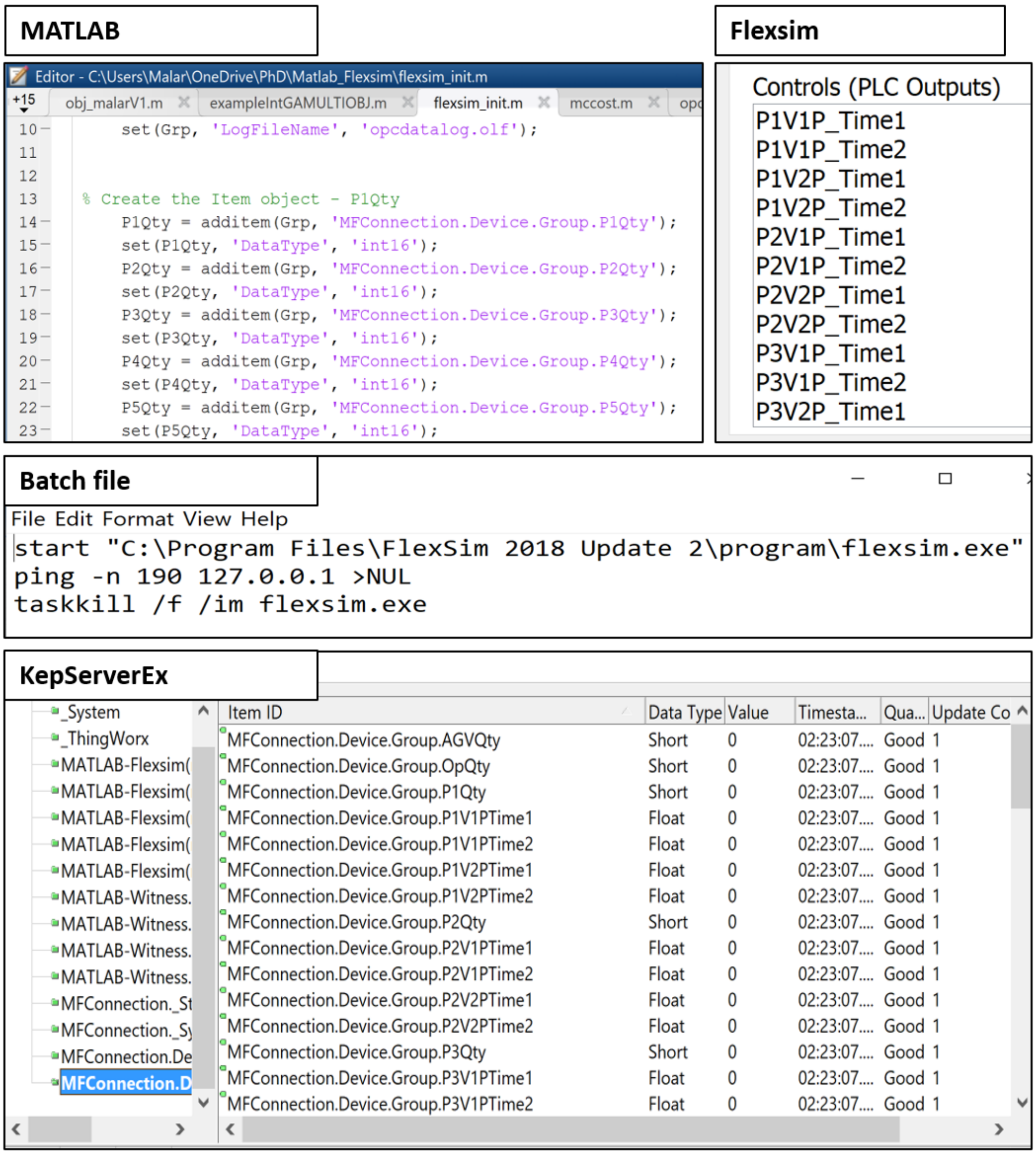


Figure 4.16: MATLAB and FlexSim integration.

elements are obtained from the common database and stored in a lookup table within MATLAB.

3. The fourth element, penalty cost, depends on the space occupation; if the total number of workstations is more than 22, which is the maximum number of available slots, then the penalty cost is considered. This evaluation of scale-up cost can be done without DES module.

4. The second objective is the throughput for which it is necessary to use DES. The *'FlexSim initialisation'* function that was described above is *'called'* to initiate the DES model. MATLAB is temporarily paused while the simulation runs and resumes on termination of DES.

5. The DES model communicates the throughput values to the the server with the help of the *'emulator'* in FlexSim. These values are read by MATLAB to calculate the second objective function.

6. A new group and two item objects are created within MATLAB for acquiring the throughput data from the server. The first item object represents the throughput value of product A and the second item object represents the throughput value for product B.

7. Using these values, the score of objective function two is obtained. In this case, both products are assumed to be equally important and hence no weights are given to throughput values. But if this is deemed necessary, it can be added to the objective function.

The simulation optimisation is achieved using a laptop with Intel Core i7 with a processor speed of 2.60GHz. **Figure 4.17** shows the trade-off solutions, also known as pareto front, obtained as a result of the multi-objective optimisation. The pareto front is a representation of the fitness evaluation scores for both objective functions. The *'x axis'* represents objective one, which is the *'scale-up cost'*, and *'y axis'* represents objective two, which is *'(1/throughput)'*. The plots to the left of the chart indicate low scale-up cost and the plots to the right indicate high scale-up cost. Each plot is a potential solution and represents a vector of the decision variable values. To ensure the validity of the non-dominated solutions, for each objective function, at the end of the generation, the best fitness score was identified and plotted in charts shown in **Figures 4.18** and **4.19**. From the plots, for the first few generations, the fitness scores are very high and as the optimisation progresses, better solutions are identified. This is clearly represented in the plots for both objectives. This ensures that the solutions on the pareto front have good fitness scores for both objectives.

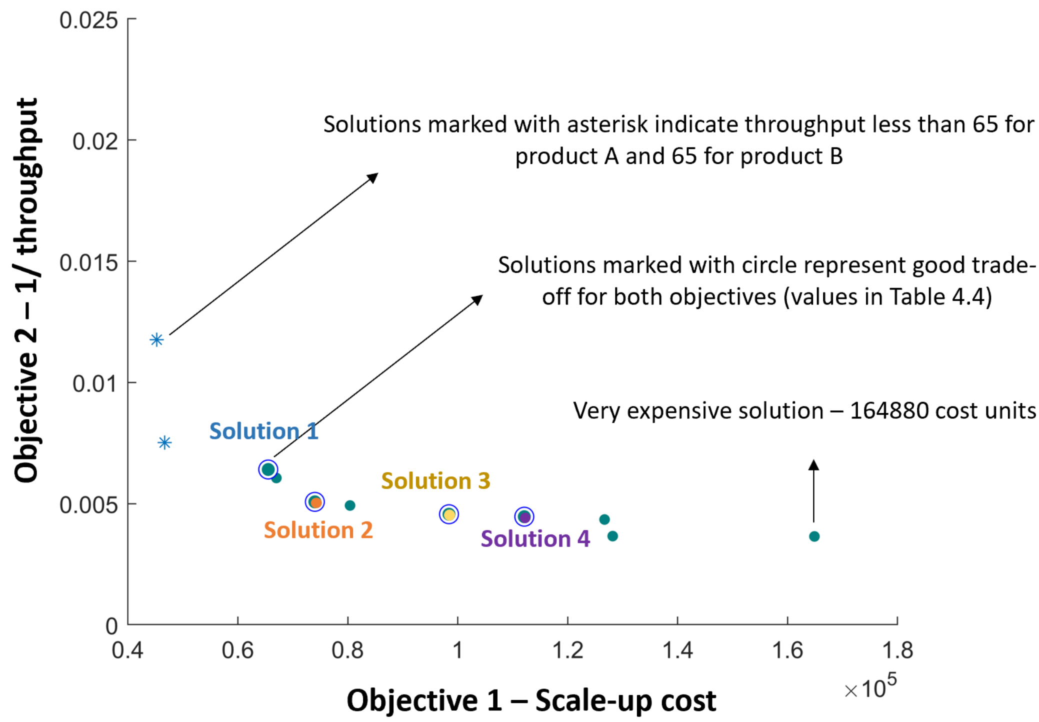


Figure 4.17: Non-dominated solutions for the battery module assembly case study.

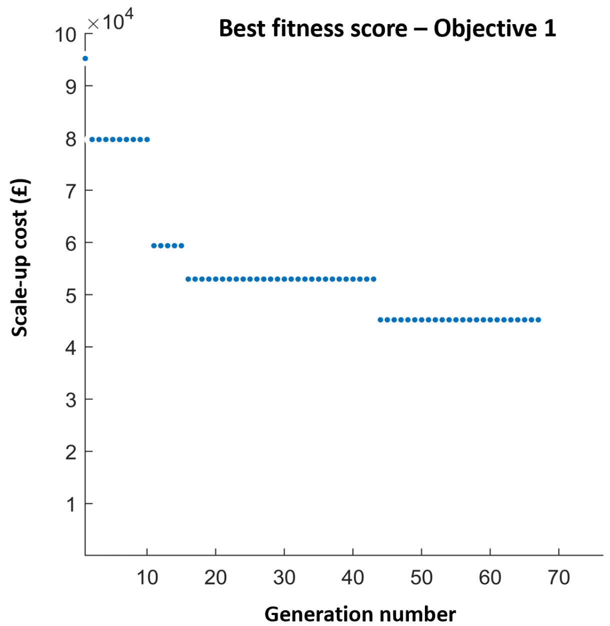


Figure 4.18: Best fitness scores for objective one.

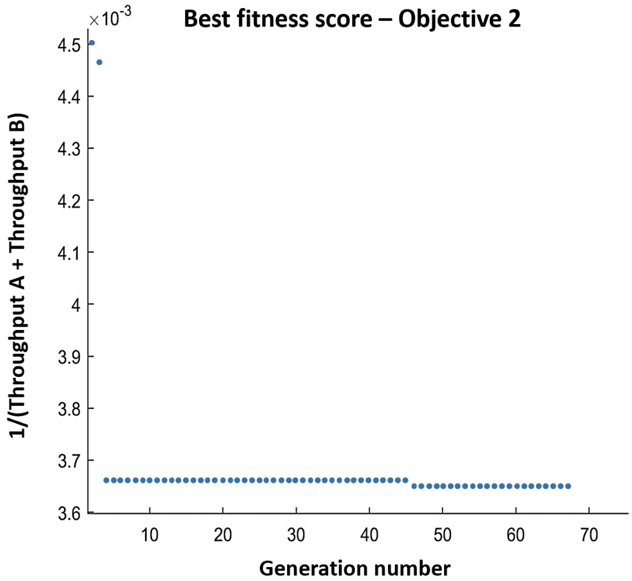


Figure 4.19: Best fitness scores for objective two.

Another plot that is used to monitor the simulation optimisation is the pareto spread. The pareto spread is an important criteria for stopping the simulation optimisation. As the simulation optimisation converges, the change in the average spread starts to reduce. This is noticeable from the plot shown in **Figure 4.20** where the average spread across the generations is provided. From the Figure, it can be seen that the average spread for the first few generations is 1. As the optimisation starts to converge at around 60 generations, the average spread value does not vary too much. The two stopping criteria considered for this case study were the i) maximum number of generations and ii) stall generation limit. Although the maximum generation limit of 100 was not reached, the optimisation was terminated since the stall generation limit criteria was achieved, i.e. the average change in spread of pareto solutions was less than the function tolerance across fifteen generations.

Having plotted the solutions in the form of pareto front, they now need to be compared. For the purpose of comparing the alternate DES scenarios, the total simulation time in DES was considered as a constant and infinite number of products were made available for processing. Although this enables the comparison of the solutions according to certain criteria, it is difficult to compare the solutions with

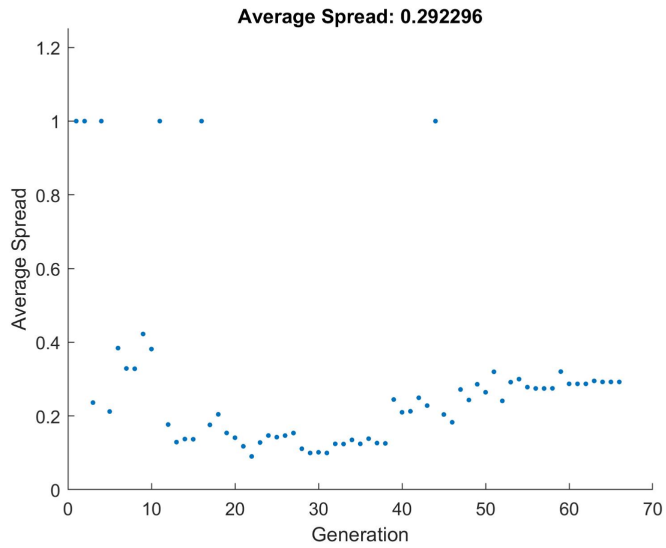


Figure 4.20: Average spread of optimisation.

regards to scheduling policies that prioritise one product over the other. Therefore, if there is need to compare the scheduling policies in DES, it might be better to allocate a finite number of products at the start of the simulation and consider the simulation time as a variable.

```
best =

   1.0e+04 *

    4.5180    0.0000

   67          1340         0.0666955          0.292296
Optimization terminated: average change in the spread of Pareto solutions less than options.FunctionTolerance.
```

Figure 4.21: Simulation optimisation convergence.

4.3.3 Decision making

For the comparison of solutions, depending on the preference of the system designer or decision maker, the best solution in their point of view needs to be selected using *'a posteriori'* approach, where the preferences are used to select a suitable solution from the considered list after the optimisation. This is a subjective procedure which involves some criteria and decision making according to those criteria. The evaluation is performed according to the *'Scale-up KPIs'* such as i) cost efficiency ($c1$/scale-up cost), where $c1$ is a constant value, ii) throughput (product A), iii) throughput (product B), iv) ease of transition, and v) compactness ($c2$/no. of slots occupied), where $c2$ is a constant value. The term *'ease of transition'* is a subjective term that is used to express the ease of transitioning, during a scale-up project, from a pilot line to a fully operational line using a particular design solution. Following this, the filtering of the optimisation results is necessary as the verification of whether the target demand is achievable by the proposed solutions is not done as part of the optimisation run.

Verification of the optimisation results

Table 4.4: Trade-off solutions selected for further analysis.

Optimisation parameters & results	Soln. 1	Soln. 2	Soln. 3	Soln. 4
No. of workstations of type 1 (x_1^1)	1	1	2	2
No. of workstations of type 2 (x_2^1)	2	2	2	3
No. of workstations of type 3 (x_3^1)	2	3	3	4
No. of workstations of type 4 (x_4^1)	1	2	2	2
No. of workstations of type 5 (x_5^1)	2	2	3	3
No. of MHUs of type 1 (x_1^2)	1	1	2	2
No. of operators of type 1 (x_1^3)	4	5	5	5
workstation configuration for station type 1 (x_1^4)	1	1	1	1
Objective 1 - Cost (units)	65520	74000	98400	112100
Objective 2 - 1/(Throughput A + Throughput B)	0.0064	0.0051	0.0046	0.0045
Throughput A (ε_1)	77	100	105	108
Throughput B (ε_2)	78	93	112	114

From **Figure 4.17**, the solutions marked with an asterisk indicate those that do not meet the required minimum throughput. The four solutions that are marked with a circle are chosen for further decision making as they provide good trade-off for both objectives considered. The decision variable values and objective function scores for the four solutions are provided in **Table 4.4**. From the table, for the purpose of validating the results of simulation optimisation, the individual throughput values for product A and product B are obtained by running the specific solutions separately in FlexSim. These values will differ, by a small margin, from those obtained from the simulation optimisation due to the use of probability distributions and pseudo random numbers. In this way, the throughput values obtained from FlexSim are verified against the results of optimisation as seen from **Table 4.5**. The comparison reveals that the 1/(Throughput A + Throughput B) values do not vary much.

Table 4.5: Verification of 1/(Throughput A + Throughput B) values.

Solution No.	Throughput from optimiser	Throughput from FlexSim run
1	0.00640	0.00641
2	0.00507	0.00519
3	0.00460	0.00459
4	0.00450	0.00447

Comparison of alternate solutions

A radar chart is provided in **Figure 4.22** to compare the considered four solutions using the indicated scale-up KPIs. The higher the value of a particular solution in the plot, the better that solution is in terms of the considered KPI. Solution one, represented in blue has the best results in terms of cost efficiency, compactness and ease of transition. Solution four, represented in purple has the best results in terms of throughput. All four solutions are capable of achieving the target throughput of 65 products of variant A and 65 products of variant B. However, solution four has far more production capacity than required. Depending on the application and scenario under consideration, the decision maker might consider implementing solution four i) if the production line is intended to be used over a long period of time and/or, ii) if the demand is expected to increase again in the future. Despite the

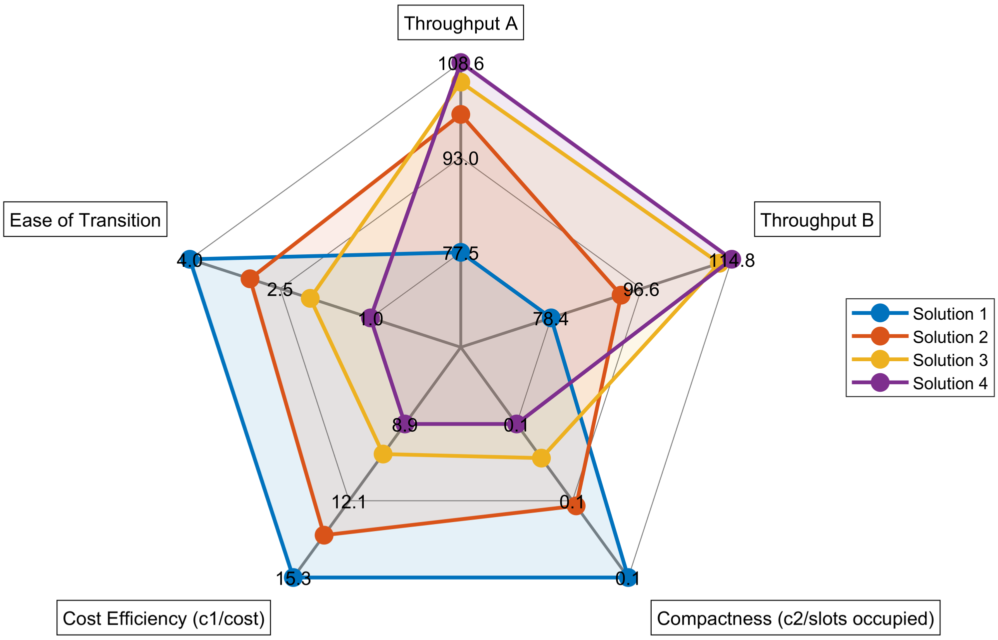


Figure 4.22: Evaluation of selected solutions with respect to scale-up KPIs.

solution being expensive and exacting a lot of effort for the transition, the buffer capacity provided by solutions three and four might be considered useful in the long run. On the other hand, the solutions one and two might be considered i) for production lines that have relatively shorter lifespan and/or, ii) for products predicted to become obsolete after a few months.

4.3.4 Summary

The results of the simulation optimisation were checked by modelling them in FlexSim, but there is no actual physical system to compare the results with. Hence, face validation with a team of experts in the field was done and according to the feedback from the discussion, the solutions were found to be reasonable and realistic. The radar chart provided a good representation and means of comparing the alternate solutions. One point to note is that the equipment that were selected during workstation configuration selection process did not have a specific price and since the primary way to obtain this was from supplier quotation, the cost values that were considered during the implementation were estimated based on the relative design and operational complexity of the considered equipment and the experience of engineers and technicians. For example, two workstation configurations were considered for station type one (cell testing and loading operations). Configuration one is relatively less expensive and comprises of three axis gantry and vacuum gripper. Configuration two is slightly more expensive because of the use of delta robot. Although the exact values of cost were not used, an estimate was done with a group of system engineers. Additionally, when the workstation configurations were selected and the optimisation was modelled, it was hypothesised that a majority of the solutions on the pareto front would comprise of configuration one which is the cheaper of the two. This is attested from **Figure 4.17**, where out of the eleven plots only the three rightmost ones consider workstation configuration two.

This concludes the demonstration of the two stage methodology on a battery module assembly case study. Examples were provided to explain the implementation of each step of the approach and tables and figures were provided to illustrate the results. The validation of the methodology was also performed and the final assembly system design solutions were evaluated in a focus group. It is now important to analyse the DDSM methodology and discuss its benefits and shortcomings.

4.4 Discussion

This section evaluates the DDSM methodology, its applicability and shortcomings. It was mentioned in Chapter one that the DDSM methodology will support scale-up transition by reducing the time, effort and cost involved in scale-up. This is also verified to determine whether the methodology is able to support the scale-up transition.

4.4.1 Evaluation of the DDSM methodology

The methodology is explored in terms of its industrial applicability and evaluated according to criteria such as *'cost, time, effort, re-usability, flexibility, applicability and traceability'*. A summary of the evaluation is provided in **Table 4.6**. In order to evaluate the methodology, a focus group was set up with participants who have experience working in the manufacturing industry and virtual modelling. The details of the discussion are provided in the following paragraphs.

Focus groups are used by market researchers to provide insights into the experiences of the participants on a subject matter such as launching a new mobile phone model and evaluating concept designs [Oates, 2000]. For the purpose of evaluating the DDSM methodology, seven participants were identified as follows: i) participant one with approximately forty years' experience working in the automotive industry, ii) participant two with four years' experience working in the manufacturing sector, iii) participant three with ten years' experience working in digital manufacturing, specifically using DES to support manufacturing industries, iv) participant four with five years' experience working the electric vehicle sector, v) participant five with thirty years' experience working in using statistical analysis to support manufacturing, vi) participant six with five years' experience working in virtual modelling, optimisation and applying artificial intelligence to support manufacturing systems, and vii) participant seven with five years' experience working in kinematic modelling. The diverse range of experience of the participants allowed the evaluation of the DDSM framework from multiple aspects. During the focus group session, face validation of the simulation optimisation results were also performed to ensure that the virtual models reflect the physical system behaviour [Banks, 1998].

The participants were provided with a thirty minutes presentation of the DDSM methodology following which various questions were asked to evaluate the methodology. The following questions were asked during the focus group session to evaluate the DDSM framework according to the considered seven criteria.

1. What are your thoughts on using the DDSM methodology to support industrial scale-up?
2. Do you think that DDSM approach can support data integration across heterogeneous modelling software such as DES, robot path planning software, operator modelling software, etc.?
3. What are your thoughts about bottom-up component modelling approach used in the methodology?
4. Do you think a decision-support system for scale-up can support the industrial scale-up projects?

5. What are your thoughts about using the methodology to reduce time-to-volume?
6. Do you think that the workstation KPIs and scale-up KPIs are comprehensive enough to represent the data used for decision-making in scale-up projects?
7. Can you provide insights into how this approach can be used in a project that you worked on?
8. How can the DDSM methodology support traceability in manufacturing systems?
9. What are your thoughts on using data from workstation models to support the simulation optimisation?

During the focus group, the participants presented their opinions and thoughts regarding the questions that were asked. The opinions and comments from the experienced participants enabled the evaluation of the methodology as follows.

Criteria 1: Time

The first evaluation criteria that is considered is the time, which in this context refers to the time-to-market and time-to-volume. In current industrial scenarios, ad-hoc procedures are put in place during scale-up planning due to various reasons such as i) need for a quick-fix without much thought, ii) carelessness, and iii) lack of proper procedures in place. As a result, the investigations regarding ergonomics, possibility of collision, human resource, space consumption, etc. are not given enough importance. Therefore, the project may get prolonged or even fail due to loss of resources. Owing to the fact that the current ad-hoc approaches lead to erroneous decisions, the time-to-market is delayed. The use of DDSM methodology, in place of the trial and error based approaches, can reduce the time taken for physical prototyping and enable faster time-to-market. This is achieved by the use of software tools that can model the various assembly line resources virtually and reduce the dependence on the experience of personnel. This subsequently contributes to reduced errors during the decision-making process.

Criteria 2: Cost

As previously explained, the ad-hoc approaches are associated with project failure, procurement of resources that might not be utilised, procurement of additional resources to replace those that were bought as a result of wrong decisions. Additionally, it is not understood whether the selected scale-up designs are actually good and efficient in terms of meeting the demand. Therefore, there is every possibility for unprecedented issues to arise due to the lack of proper investigation. To overcome this, there is the urgent need to replace the failed design or plan with anything that can do the job. The cost of making such mistakes is very expensive. Since the designs or solutions were not investigated and thoroughly compared with alternate solutions, there is the risk of them being sub-optimal and possibly more expensive than required. This leads to excessive spending that is usually unnoticed.

With the use of DDSM methodology, every solution or design that is selected can be justified at every step. This allows tight monitoring of project spending on worthwhile solutions. It is also possible to assess the new assembly system designs with respect to system KPIs to understand whether the new solutions lead to an improvement in the KPIs. This is possible due to the modelling of what-if scenarios in DES coupled with the optimisation module. Another aspect is the issue with current PLM platform having a lucrative outlook; therefore, when a PLM suite needs to be implemented in industries, it demands expensive transformation, change and adaptation. As an alternative, industries can benefit from approaches such as DDSM that provide framework for communication among several tools that are not part of the same platform or language. In this way, it saves a lot of cost for industries and gives them the freedom to choose the software that they need. It is also possible to create their own software to perform the kinematic modelling, DES and optimisation and still achieve interoperability at a relatively cheaper cost.

Criteria 3: Effort

In any manufacturing system, during planning stages, there is usually a number of possible solutions that need to be considered. This becomes all the more difficult with the increase in the complexity of the system. This makes it cognitively complex for the human brain to choose a particular solution. Without proper decision support, the planning stage is cognitively complex and can inadvertently result in errors. With the additional burden of project time constraints, the effort that engineers and managers need to put into the project increases. The whole procedure of investigation, followed by procurement and commissioning is a daunting one. Therefore, the DDSM methodology helps to reduce the burden from the shoulders of managers and engineers. This is achieved by creating libraries in the kinematic modelling software, DES and ontology editor. The elements in the library only need to created once and they are re-usable.

The decision support provided in Stage two for the assembly system design selection can speed up the decision making process while still maintaining the quality of the solutions. It is also possibility to use the approach for product variant analysis. The mapping of product, process and resource in Protégé also allows change control since the links and associations are embedded in the model. Therefore, the impact of the change in a workpiece-related parameter on the process and system resource can be identified automatically.

Criteria 4: Reusability

The use of various libraries within kinematic model, creation of model templates in DES and creation of equipment catalog within the ontology editor allows the encapsulation of data from the assembly line at various levels of granularity. As a result, when new concepts need to be developed in the planning stages, the availability of libraries and model templates saves a lot of time that can be redirected to productive decision making and analyses. This further feeds into reducing the effort and time-to-market. The concept of one time modelling and multiple reuse is possible due to the current era of digitalisation. Hence, it is important to leverage it carefully to gain maximum benefit. In DDSM, the DES model is parametric due to the need to connect to the optimisation module. Some of the considered parameters include the number of workstations, operators and transporters. Since a majority of *'what-if'* scenario analyses in DES make use of the mentioned parameters, there is possibility to adopt this model as a template for various applications. Further work needs to be done in this area and this will be discussed in Chapter five.

Table 4.6: Evaluation of DDSM methodology.

Evaluation criteria	Assessment of DDSM methodology
Time	reduces time-to-market and time-to-volume
	virtual validation of concepts reduces the time spent on physical prototyping
	reduces human errors
Cost	reduces risk of choosing expensive sub-optimal solutions
	reduces risk of project failure
	provides alternate solution to product lifecycle management suites
Effort	provides decision support for cognitively complex design solutions
	pre-defined libraries reduce effort involved in virtual model creation
Reusability	use of parametric models supports reusability
	data encapsulation in virtual models supports planning stages
Extendability	the methodology could be extended by addition of other software
Traceability	use of common database with IDs enables traceability
	use of digital twins enables performing quality checks at every stage
Applicability	applicable to industries that envision digital transformation
	decision support using virtual models drives the digital transformation

Criteria 5: Extendability

The term *'extendability'* in this context refers to ability to extend the methodology by adding other software for various applications such as robot path planning, mobile robot routing, etc. Extendability is an important criteria during the planning stages. The DDSM methodology is found to be extendable due to the parametric nature of the models. The method is not limited to a particular phase of the lifecycle and can be used whenever scale-up planning is required. Although the type of data that is used might vary depending on the scale-up is pre-operational or operational phase, the core of the method remains the same. Since the DDSM provides a framework for supporting interoperability of heterogeneous software, there is no limitation on how many different software can be added to the methodology. It is possible to link the methodology with software that specialise in analysing ergonomics, AGV fleet manager, robot path planning and modelling software, human resource analysis software, etc., to improve the accuracy and applicability to various situations. Moreover, the primary idea is to have an open source solution that considers the interoperability of various key software used for scale-up transition.

Criteria 6: Traceability and quality

Traceability and quality play an important role in the case of certain hazardous products and materials that can have implications on health and safety. This is indeed true for any assembly related to the electric vehicle powertrain. In such situations, the use of a common repository like the database along with unique identifiers for workpiece, equipment, processes and configurations, makes it possible to trace each and every step of the product and associated resources from the start to finish of the assembly. Therefore, it is possible to ensure that the quality guidelines are met before progressing to the next step. Also, safety critical products can be referenced to the machine or operator who performed the assembly process, in addition to the monitoring of operator skill level and certifications. The use of a common database ensures that all the relevant data are available in a single space and this can benefit the whole supply chain. There is possibility to extend the research in this area and this will be discussed in Chapter five.

Criteria 7: Applicability

The methodology is built upon the use of digital tools and extends the capabilities of digital manufacturing. In this era of digital transformation where a number of companies are just venturing into the new field, an approach that supports the digital transformation by enabling the communication between heterogeneous software is considered applicable. It is important to note that this would not have been the case few years ago. The methodology is generic enough to be applied to any stage of the life-cycle where scale-up might be necessary. It can also be applied to any type of assembly system that can be decomposed as component, workstation, pilot line, production line and factory as mentioned in section 1.3.

4.4.2 Evaluation with related works

The DDSM approach is closely related to the work done by Ghani [Ghani, 2013]. The research work done on the integration between Kinematic modelling software and DES proposed by Ghani et al. [Ghani, 2013] is adopted for the DDSM approach to support scale-up decision making. However, in DDSM, the kinematic modelling software is enriched with knowledge representation using the ontology editor. Moreover, the DES model is coupled with an optimisation algorithm to support the scale-up decision making. The other related works include the research done by Michalos et al. [Michalos et al., 2015] and [Manzini et al., 2018]. Both these works focus on the system configuration and design problem. While Michalos et al. [Michalos et al., 2015] support the robotic workstations using a two-stage approach combining analytical method and virtual modelling, Manzini et al. [Manzini et al., 2018] support the modular assembly systems using a knowledge-based tool. In DDSM, however, both station and assembly line configuration and design selection are supported with the help of virtual modelling tools. A comparison of the related works is provided in **Table 4.7**.

4.4.3 Limitations

The evaluation of the methodology helps understanding the shortcomings that need to be addressed. They are discussed as follows:

Kinematic model building and ontology editor capabilities

Specific to the Workstation Configuration Selector, the use of ontology editor is intended to retrieve the equipment that perform a specific operation. This information is used to generate the workstation configurations that are validated in a kinematic modelling software. This process might be time consuming if the kinematic models have to be built from scratch. To reduce the time taken for modelling in the kinematic software, it might be possible to use the ontology model to calculate the workstation process time for the various configurations. In this way, the configurations that do not meet the required process time can be eliminated and do not need to be modelled within kinematic modelling software. However, the creation of such an ontology model is a complex one.

Table 4.7: Comparison of DDSM methodology with related works.

	[Ghani, 2013]	[Michalos et al., 2015]	[Manzini et al., 2018]	DDSM (presented work)
Research focus	Integration of DES and kinematic model	Production line configuration problem	System design and reconfiguration problem	Scale-up decision support
Application area	Reconfigurable assembly systems	Robotic workstations	Modular assembly systems	Assembly systems
Station configuration	Kinematic modelling	Analytical method	Knowledge-based cell configuration tool	Knowledge-based kinematic modelling
Line configuration	DES modelling	Virtual modelling approach	Knowledge-based system configuration tool	Simulation-based optimisation

Ontology editor and query language

In this research, the ontology editor that is used is Protégé and the query language is SQWRL. Accordingly, the query design is specific to the SQWRL language. Therefore, if another ontology editor or query language is used, the execution will remain same but the structure of the query might look different. More work needs to be done on this to ensure that the methodology is not limited to Protégé software.

Applicability to different task types

For the equipment selection, only five task types were considered. In reality, there are more tasks such as testing that were not considered in the research. Moreover, the case study only considered the move and hold/release tasks. The application of the methodology for the welding, joining and transportation tasks are not discussed in detail.

Approach to simulation optimisation

Regarding the SCS, the use of simulation optimisation with approximately around 15 - 20 decision variables adversely impacts the computer's memory. This limits the maximum number of generations that can be run. If the considered system is very complex, then there is the risk that the optimisation will not converge within the defined number of generations. This is further aggravated by the use of OPC UA server for real-time data communication since the creation of item objects in MATLAB adversely affects the memory consumption.

Another issue was the quality of throughput data obtained from FlexSim. The throughput value could not be read in some situations due to connectivity issues and subsequently resulted in erroneous values. For this purpose, the fitness evaluation code was modified to read the throughput value from the server multiple times until the data quality was ensured. In some iterations, several loops of this code were run until the actual throughput value was obtained which took an additional 30 seconds to a minute per iteration of the optimisation. Moreover, the triggering of FlexSim from within MATLAB was done for every single iteration. This activity took more time than the simulation run itself. The mentioned issues are software-specific and the situation might be different with the use of other software for DES, optimisation and connectivity.

Incorporating warehouse and logistics

DDSM methodology has been tested for battery module case study, but its applicability to other assembly lines along with the consideration of warehouse and logistics is not done. The optimisation procedure is limited in its application; there is possibility of increasing its capabilities by incorporating layout optimisation and operational policy analyses. The author believes that it will be worthwhile to pursue research in this direction and the relevant discussions are provided in the next chapter.

Data collection

Since the methodology uses virtual modelling, the input data plays a very important role. This underscores the importance of data acquisition systems. However, the methodology has not considered this in much detail.

4.5 Chapter summary

The main aim of this chapter was to demonstrate the DDSM methodology with a case study. The methodology is divided into the workstation configuration selection stage and the system configuration selection stage. A step-by-step implementation was provided considering the battery module assembly test case. The results of the analysis were presented in the form of radar chart and the comparison of alternate designs was facilitated using the scale-up KPIs. The benefits of adopting this approach in an industrial setting was discussed and evaluated using a diverse set of criteria. The chapter concludes by listing the shortcomings of the DDSM methodology. The next chapter discusses the future activities that could stem from the DDSM research work and reviews the original hypothesis, research question and objectives that were proposed in Chapter one.

Chapter 5

Future work and Conclusion

This concluding chapter of the thesis starts with the assessment of whether the three main research objectives proposed in Chapter one are fulfilled. This is followed by a review of the research question and research hypothesis. The key contributions of the research are discussed along with the proposals for future research activities that could complement the DDSM methodology.

5.1 Revisiting the research objectives

5.1.1 Objective one

Identify the data from the physical system/shop floor that are required by digital simulation tools, namely kinematic modelling tool and DES, which are used for modelling workstations and production lines, respectively.

In order to fulfill this objective, literature on input data for virtual models and data acquisition was reviewed. From the study, the data utilised in the kinematic modelling software and DES were categorised and tabulated. The data sources, data format and the importance of the data were investigated. This played a vital role in determining the type of data integration for the DDSM methodology and understanding the dataset from the kinematic modelling tool that could enrich the DES models.

5.1.2 Objective two

To propose a robust framework for multi-domain data integration of software at two different levels of granularity, namely the workstation and system level, to identify potential workstation and system configurations that can accommodate the increased capacity following scale-up.

This objective plays a vital role in answering the research question and identifies a way to leverage the data integration between the kinematic model and DES model to provide a decision support framework for scale-up planning. This objective was achieved by proposing the Data-Driven Scale-up Model (DDSM) that is capable of utilising the data from different software at different granularity levels. The Stage one of the methodology, known as Workstation Configuration Selector (WCS), employs the ontology editor and the kinematic modelling software to identify potential workstation configurations. The Stage two, known as the System Configuration Selector (SCS), employs DES and optimisation modules to identify potential system configurations while having access to the workstation level data.

5.1.3 Objective three

To demonstrate the application of the proposed methodology to support the transition from low to high volume production in a pilot line case study.

The framework was demonstrated in a battery module assembly line for a scale-up scenario with the new demand being approximately three times the current demand. The DDSM methodology was evaluated and found beneficial for industries to provide a systematic method to move through the scale-up phase. Owing to the budget constraints, the validation of the new assembly line design and configuration by building a real production system is difficult. For this reason, i) a virtual model of the solution in FlexSim along with the kinematic model in vueOne, and ii) a radar chart comparing the solutions were presented to industrial experts.

5.1.4 Review of research question and hypothesis

'How can the data integration and interoperability between kinematic and DES models for decision-making regarding the assembly system design during scale-up planning phase be achieved in a seamless way?'.

The completion of the three research objectives sheds light on the answer to the research question. The decision-support framework, namely DDSM, built upon integration of heterogeneous software models is proposed in Chapter three. Through the framework, the data integration of kinematic and DES models was explored and the benefits of enriching the DES models with workstation data were highlighted. The benefits on assembly system design selection was also demonstrated. The common database serves as a repository for system lifecycle data and ensures data integrity. In this way, the proposed framework answers the research question.

The hypothesis that was introduced in Chapter one states that employing the DDSM framework reduces time to volume and supports decision making for selecting good assembly system design. To check this, a focus group with field experts was established and were presented with the case study and potential system designs. They agreed that using the data-driven approach enables selection of optimum solutions from the design space and reduces the chances of project failure. Additionally, they concluded that the time taken to reach the desired volume is also shortened due to data organisation using common repository and using virtual models for concept planning. A more detailed evaluation of the methodology was provided in Chapter 4.

5.2 Contributions

The thesis presents four key contributions that are highlighted below.

1. This framework addresses the current issues faced due to experience-based scale-up in industries that results in extended scale-up duration, project failure and expensive design solutions. There is lack of a robust framework to support up-scaling of assembly lines. ***To overcome this, a holistic data-driven decision support framework that reduces i) development and changeover***

time, ii) scale-up cost, and iii) effort involved, is presented in section 3.1 and evaluated in section 4.4.

2. ***The DDSM methodology employs simulation optimisation for system configuration selection and kinematic model enhanced by knowledge mapping for workstation configuration selection. This supports scale-up decision making and is a novel contribution to literature.*** Through the approach, it is possible to screen the design space at varying levels of granularity for potential solutions.

3. The issue of modelling infeasible design solutions in DES is one that is not explored in detail. ***This is tackled using a hybrid approach where the integration of knowledge mapping and kinematic modelling software improves the accuracy of DES models, subsequently providing better design solutions. The novelty is highlighted by comparing with relevant works in section 4.4.2.*** The encapsulation of data at the lower-level models improves accuracy of DES models.

4. There are a number of different ontology models that exist in literature. However, an ontology model for scale-up equipment selection is lacking. ***For this purpose, a novel Product, Process, Resource and Resource attribute ontology model for assembly system configuration selection is proposed in section 3.3.2. The ontology structure and query design are tailored for the scale-up equipment selection problem.***

5.3 Future research directions

5.3.1 Product variant analysis

The DDSM methodology mainly focuses on production volume increase, however, in industries, the volume increase might be accompanied by new product introduction. In such situations, it is important to understand if the new assembly line configuration can accommodate not only the increased demand but also the new product variant. Using the ontology editor and kinematic models, it is possible to do product variant analysis. [Chinnathai, Günther, Ahmad, Stocker, Richter, Schreiner, Vera, Reinhart and Harrison, 2017] provides a detailed approach for product variant analysis within kinematic modelling software with the help of the mappings between the product dimensions and resource features. A future research direction could be to incorporate the approach provided in into the DDSM methodology to increase its capabilities.

5.3.2 CAD annotations

In a number of cases, the CAD models of the workpiece, resource and equipment are not easily transferable to the other software in the system lifecycle. Even if it was possible to transfer the model, it would be associated with loss of data when the native CAD is converted to a another format. However, it is possible to ensure that the important data such as the dimensions of the product, properties of an equipment, etc., are made transferable by annotating them. This allows for data reuse and reduces the time taken for the modelling and creating component library within the kinematic modelling software.

At the moment, the creation of component kinematic models is time consuming due to the manual input of product and equipment data. However, more research work needs to be done in automatically updating the kinematic model with the relevant process and resource parameters. This can reduce human errors and save a lot of time.

5.3.3 Multi-criteria decision making

Decision making plays a very important role in the methodology, therefore, more emphasis should be given to this activity. In the current version of decision making, the scale-up KPIs are used to score and compare the alternate scenarios on a radar chart. However, there are more systematic approaches for decision making and hence, the results of the pareto front could be coupled with a Multi-Criteria Decision Making (MCDM) techniques such as Analytical Hierarchy Process(AHP) or Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) to get better results. Additionally, a more comprehensive list of scale-up KPIs should also be created.

5.3.4 Improving the DES and optimisation module

The use of DES and optimisation module in the methodology allows the analysis of data to arrive at various conclusions. Although in this methodology, GA is the only algorithm employed, there are several other algorithms and methods that need to be explored and applied in the methodology. In this regard, the optimisation could be extended to perform layout and operational analysis as well. It is also possible to monitor the assembly line KPIs in DES and connect it to dashboards and visualisation platforms for better representation of the results and performance measures.

5.3.5 Decision support UI

In the DDSM, the core areas of interaction between the system engineers and the methodology is when decisions need to be made. There is potential to improve the proposed decision support using an intelligent user interface that would get user inputs such as process parameters, preferences, etc., that will be used in DDSM framework to propose assembly system design configurations such that the workload on the system engineer is reduced. For this purpose, the areas of the methodology that need human interaction can be improved with better Graphical User Interface.

5.3.6 Impact of software change

In the current version of the DDSM, the following software are employed: FlexSim, MATLAB, Protégé and vueOne. These software can be replaced with several other software which have the same functionality. Moreover, industries might develop their own domestic software as well. Therefore, further work needs to be done to identify the impact of changing a particular software with alternate ones in the DDSM methodology.

5.3.7 Use case for operational phase scale-up

In this research, the DDSM was demonstrated with a battery module case study in the early implementation phase known as pre-operational scale-up. However, a demonstration of the methodology for the operational phase scenario was not provided. Further work needs to be done in this direction.

5.4 Conclusion

The primary aim of this research work was to propose a methodology that provides scale-up decision support using heterogeneous software packages. To achieve this, firstly, an in-depth review of existing literature was done to investigate the relevant works and understand the research gaps. The current industrial practices for scale-up and input data for virtual models were also explored. Following the investigation, an approach for selecting workstation and system configurations for scale-up was proposed. The approach, named as DDSM, is a two-stage framework that screens the design space at two different levels of granularity.

The first stage of the methodology, known as Workstation Configuration Selector (WCS), utilises the knowledge representation and kinematic modelling modules to select workstation designs that meet the process requirements. The second stage of the methodology, known as the System Configuration Selector (SCS), utilises the DES and optimisation modules to select the system designs that meet the required production volume. Additionally, stage one enriches the selection process in stage two by providing necessary workstation data to the DES module. This is achieved using a common database. The results of the selection process were represented using a radar chart and the methodology was evaluated using certain criteria such as the time, cost, effort, reusability, traceability and applicability. Finally, the future research directions emanating from the DDSM methodology were discussed.

Appendix A

Journal publication

A Novel Data-driven Approach to Support Decision-Making during Production Scale-up of Assembly Systems

Malarvizhi Kaniappan Chinnathai[a,*], Bugra Alkan[b], Robert Harrison[a]

[a]*Warwick Manufacturing Group, University of Warwick, Coventry, CV4 7AL, UK*
[b]*School of Engineering, London South Bank University, London, SE1 0AA, UK*

Abstract

In today's manufacturing settings, a sudden increase in the customer demand may enforce manufacturers to alter their manufacturing systems either by adding new resources or changing the layout within a restricted time frame. Without an appropriate strategy to handle this transition to higher volume, manufacturers risk losing their market competitiveness. The subjective experience-based ad-hoc procedures existing in the industrial domain are insufficient to support the transition to a higher volume, thereby necessitating a new approach where the scale-up can be realised in a timely, systematic manner. This research study aims to fulfill this gap by proposing a novel Data-Driven Scale-up Model, known as DDSM, that builds upon kinematic and Discrete-Event Simulation (DES) models. These models are further enhanced by historical production data and knowledge representation techniques. The DDSM approach identifies the near-optimal production system configurations that meet the new customer demand using an iterative design process across two distinct levels, namely the workstation and system levels. At the workstation level, a set of potential workstation configurations are identified by utilising the knowledge mapping between product, process, resource and resource attribute domains. Workstation design data of selected configurations are streamlined into a common data model that is accessed at the system level where DES software and a multi-objective Genetic Algorithm (GA) are used to support decision-making activities by identifying potential system configurations that provide optimum scale-up Key Performance Indicators (KPIs). For the optimisation study, two conflicting objectives: scale-up cost and production throughput are considered. The approach is employed in a battery module assembly pilot line that requires structural modifications to meet the surge in the demand of electric vehicle powertrains. The pilot line is located at the Warwick Manufacturing Group, University of Warwick, where the production data is captured to initiate and validate the workstation models. Conclusively, it is ascertained by experts that the approach is found useful to support the selection of suitable system configuration and design with significant savings in time, cost and effort.

Keywords: Manufacturing systems, production planning, scale-up, demand amplification, demand uncertainty, data-driven method, discrete-event simulation, DES, multi-objective optimisation, evolutionary optimisation algorithm, genetic algorithm, GA, kinematic modelling.

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]	[redacted]		[redacted]			[redacted]	
	[redacted]						
[redacted]	[redacted]			[redacted]		[redacted]	
[redacted]		[redacted]	[redacted]				[redacted]
[redacted]		[redacted]	[redacted]				[redacted]
[redacted]		[redacted]	[redacted]				[redacted]
[redacted]	[redacted]				[redacted]		[redacted]
[redacted]	[redacted]		[redacted]				[redacted]
[redacted]		[redacted]	[redacted]				[redacted]
[redacted]		[redacted]	[redacted]				[redacted]
[redacted]		[redacted]	[redacted]				[redacted]
[redacted]		[redacted]	[redacted]				[redacted]
[redacted]		[redacted]	[redacted]				[redacted]

[redacted]

[redacted]	[redacted]		[redacted]			[redacted]				
	[redacted]									
[redacted]	[redacted]		[redacted]			[redacted]				
[redacted]	[redacted]				[redacted]		[redacted]		[redacted]	
[redacted]	[redacted]				[redacted]				[redacted]	
[redacted]		[redacted]	[redacted]						[redacted]	
[redacted]		[redacted]	[redacted]					[redacted]	[redacted]	[redacted]
[redacted]	[redacted]				[redacted]				[redacted]	
[redacted]		[redacted]	[redacted]					[redacted]		[redacted]
[redacted]	[redacted]		[redacted]					[redacted]		
[redacted]	[redacted]				[redacted]					
[redacted]	[redacted]			[redacted]		[redacted]				

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]

[redacted]

[redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]		
[redacted]	[redacted]	[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]	
[redacted]	[redacted]

[redacted]

[redacted]	
[redacted]	[redacted]

[redacted]

Process 2 – Kinematic model of cell loading

Process 2 – Real counterpart

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]		[redacted]	
		[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]		[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

Appendix B

Conference publications

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 60 (2017) 74 – 79

www.elsevier.com/locate/procedia

27th CIRP Design 2017

Convertibility evaluation of automated assembly system designs for high variety production

Malarvizhi Kaniappan Chinnathai*, Bugra Alkan, Robert Harrison

Automation Systems Group, WMG, University of Warwick, CV4 7AL, Coventry, West Midlands, UK

* Corresponding author. Tel.: +44 7831001684; *E-mail address:* M.Kaniappan-Chinnathai.1@warwick.ac.uk.

Abstract

The recent advancements in technology and the high volatility in automotive market compel industries to design their production systems to offer the required product variety. Although, paradigms such as reconfigurable modular designs, changeable manufacturing, holonic and agent based systems are widely discussed to satisfy the need for product variety management, it is essential to practically assess the initial design at a finer level of granularity, so that those designs deemed to lack necessary features can be flagged and optimised. In this research, convertibility expresses the ability of a system to change to accommodate product variety. The objective of this research is to evaluate the system design and quantify its responsiveness to change for product variety. To achieve this, automated assembly systems are decomposed into their constituent components followed by an evaluation of their contribution to the system's ability to change. In a similar manner, the system layout is analysed and the measures are expressed as a function of the layout and equipment convertibility. The results emphasize the issues with the considered layout configuration and system equipment. The proposed approach is demonstrated through the conceptual design of battery module assembly system, and the benefits of the model are elucidated.

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference

Keywords: Assembly systems; product variety; convertibility; design evaluation.

1. Introduction

Due to the increasing importance to satisfy customer needs, there has been a shift from mass production to mass customisation in the automotive market [1]. In order to survive in this competitive, turbulent and highly volatile market, enterprises have to employ new practices and strategies that can effectively accommodate high variety production to realise the advantage of mass customisation [2]. Thus, the concept of product variety management has gained significant importance within the last two decades. A key enabler for this is considered to be convertibility which is defined by [3] as "*the ability to easily transform the functionality of existing systems and machines to suit new production requirements*". To realise this, several approaches have been proposed for designing systems with the ability to handle the increasing product variety and fluctuating volume. However, unless an evaluation of the systems designed based on these approaches is performed, it is difficult to ascertain their capability to manage product variety. Hence, it is important to assess the system's responsiveness and ability to adapt to change, especially in the early design stages, since poor initial design increases the effort and time spent during redesign later in the design and engineering process[4]. Hence, this paper proposes a novel design support mechanism which can assess the concept designs of automated assembly systems, in an industrial-friendly way, for their readiness to change to a new configuration.

2. Literature review

Over the past few years, the domain of product variety management and flexible systems have received lot of attention. As a result, a number of models and methods to evaluate the flexibility of system have been researched in literature [5], [6]. However, there is limited research in the field of convertibility, which is considered as one of the characteristics of reconfigurability. Although convertibility is associated with product variety management, it is difficult to

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference
doi:10.1016/j.procir.2017.01.005

hypothesize a convertibility measurement using existing flexibility assessment models. Additionally, they need significant amount of data which is unavailable at the conceptual phase. Therefore, the literature study is limited to research on reconfigurability and convertibility evaluation.

Nomenclature

C_S	System convertibility
C_E	Equipment convertibility
C_C	Component convertibility
w_E	Weight for equipment convertibility
w_L	Weight for layout convertibility
C_L	Layout convertibility
$C_{SS,k}$	Convertibility of sub-system *k* (equipment level)
N	Number of sub-systems
M_k	Number of components in sub-system *k*
$f_{h,i}$	Hardware convertibility factor
$f_{s,j}$	Software convertibility factor
n	Number of hardware convertibility factors
m	Number of software convertibility factors
x	is 2 for controlled and 1 for uncontrolled components
N_k	Number of sub-systems, excluding sub-system *k*, shut down when sub-system *k* is under conversion
N_F	Total number of part flow connections, excluding input and output
N_{AWS}	Number of assembly workstations
N_R	Minimum number of replicated stations
L_A	Autonomy index
L_C	Connectivity index
L_R	Replication index

In the domain of reconfigurability, an approach for assessing the re-configurability of distributed manufacturing systems was proposed in [7]. In a similar study, Hasan et al. [8] investigated the re-configurability of machines through Multi-Attribute Utility Theory and Power function approximation. In the study, the re-configurability of machine configurations was evaluated based on machine attributes such as possible number of configurations, operational capability, effort required to reconfigure and production capacity of the machine. Farid [9] synthesised a re-configuration measure based on axiomatic design theory and design structure matrix to derive composite reconfiguration evaluation. A measure of the system's convertibility was formulated by the summation of the transportation and transformation convertibility in the work. Convertibility was measured in three different domains by [10], namely: configuration, machine, and material handling. The configuration convertibility was quantitatively evaluated with variables such as routing connections, replicated machines, and increment of change. Machine and material handling convertibility were intuitively scored. The combined score of the three domains provides a multi-dimensional convertibility value which is a representative of the system. This evaluation model was further improved by an adaptation to mixed-model assembly lines by [11], wherein a novel product family convertibility analysis was introduced.

An approach to measure the machine reconfigurability and operational capability was proposed by [12] and the possible number of possible machine configurations and the effort involved in changing them were identified. A metric called 'reconfiguration smoothness' was measured based on the cost, effort and time spent in system reconfiguration by [13]. Various aspects of change involved at machine level, system level and market level were considered. Each was expressed as a function of either the capabilities, or the machines added, removed or adjusted in the system. Ahmad et al. [14] describe an approach to evaluate the reconfigurability of an hydrogen fuel cell assembly system and analyse its suitability to the product. The approach intuitively measures a Reconfigurable Assembly System (RAS) for its conformity to the various aspects of reconfigurability including convertibility.

From the above-mentioned studies, it is observed that there is lack of sufficient research on the evaluation of convertibility of assembly systems in the concept phase that can assist in system redesign to achieve an optimum level of flexibility. To fulfil this gap, a novel evaluation model to assess the assembly system, for product variety at the concept stage, is proposed. The model can flag the system components at various levels of hierarchy that will later help formulate a multi-criteria redesign policy that can guide the designer to achieve a system capable of managing variety.

3. Methodology

The scope of this research is defined around the analysis of automated assembly system design convertibility based on its equipment structures and layout (Fig. 1). In this approach, an industrial assembly system is defined as a hierarchical network consisting of assembly workstations (AWS), connected through material handling units (MHU). System convertibility C_S is defined as an average of equipment convertibility C_E, and layout convertibility C_L and calculated by Eq. 1., where in order to provide decision-making flexibility in system assessment, w_E and w_L represent the weights for C_E and C_L respectively

$$C_S = w_E C_E + w_L C_L \tag{1}$$

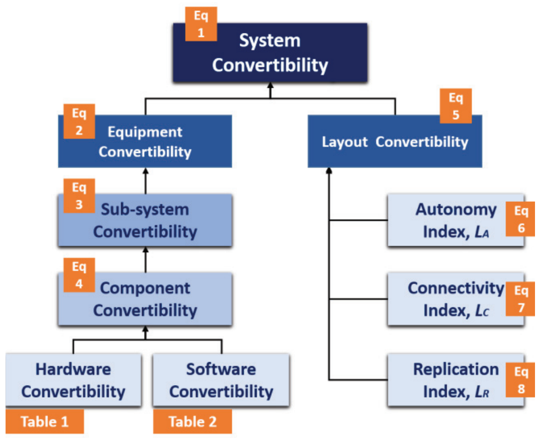


Fig. 1 The proposed methodology.

3.1. Equipment convertibility

Equipment convertibility C_E is the ability of equipment in a system to be changed or adjusted, by the addition, removal or adjustment of its constituent components. In this context, C_E is defined as a function of the convertibility of each assembly sub-system, N represents the number of sub-systems, C_{SS}, representing either workstations or material handling units, and is calculated as follows (Eq. 2).

$$C_E = \frac{\sum_{k=1}^{N} C_{SS,k}}{N} \quad (2)$$

In this study, a sub-system is assumed to be composed of a set of re-usable automation components (*e.g*. rotary table, clamp, gripper, *etc*.). M_k is defined as the number of components in sub-system k, C_{SS} is defined as the average component convertibility within the sub-system and it is calculated by Eq. 3.

$$C_{SS} = \frac{\sum_{l=1}^{M_k} C_{C,l}}{M_k} \quad (3)$$

A component is defined as the basic unit of a sub-system which at a finer level is composed of elements [15], and is capable of functioning either autonomously and/or integrated with other components to perform its desired function [16]. In this context, two types of classifications of component have been made; *i.e.* control and function. The classification based on control requirements categories components into two groups, *i.e.* controlled or non-controlled. Components that do not have control logic, and can be assessed only from the hardware perspective are denoted as non-controlled components (*e.g.* passive fixtures). On the other hand, controlled components can be actuated and hence are associated with control logic (e.g. active fixtures). Therefore, they must be assessed on both hardware and software domains. In function-based classification, the components are classified into five types *i.e.* motion, holding, joining, transport, and feeding components. By adapting the coding approach proposed in [17], the component convertibility, C_C, is calculated using the following equation, where ‘n’ and ‘m’ represent the number of hardware and software convertibility factors respectively.

$$C_C = \frac{1}{x}\left(\frac{\sum_i^n f_{h,i}^2}{n} + \frac{\sum_j^m f_{s,j}^2}{m}\right) \quad (4)$$

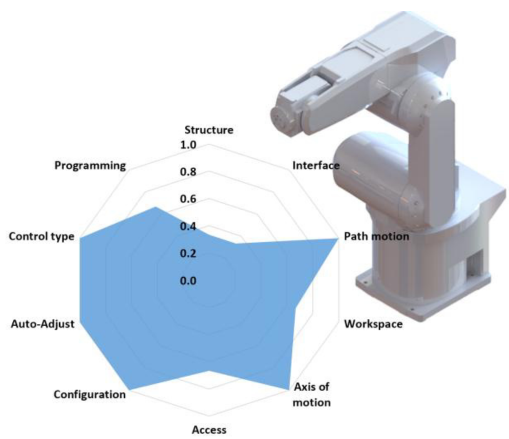
In this context, the hardware convertibility factors are calculated for all components regardless of their control behaviour, however the factors vary depending on their functions. Irrespective of the function of the component, the software convertibility factors are generic and calculated only for controlled components. Adapted from Table 1 and Table 2 represent hardware and software component convertibility factors respectively, and it is assumed in this study that these factors impact the system convertibility. However, components in an assembly system which are used for measurement or inspection, *e.g.* sensors, test gauge etc. and components which

Table 1. Hardware convertibility scores.

	Function	*i*	Criteria	0	0.333	0.667	1
1	Motion	1	Structure	-	Fixed	-	Modular
		2	Interface	Static/irremovable	Complex/non-standard	-	Simple/standard
		3	Path motion	Fixed	-	-	Variable
		4	Workspace	-	Tight	Appropriate	Large
		5	Axis of motion	-	1-2	3-4	5-6+
2	Holding	1	Structure	-	Fixed	Modular	Reconfigurable
		2	Interface	Static/irremovable	Complex/non-standard	-	Simple/standard
		3	DOF	-	0	1-2	3+
3	Transport	1	Structure	-	Fixed	-	Modular/extendable
		2	Interface	Static/irremovable	Complex/non-standard	-	Simple/standard
		3	Direction	-	Unidirectional	Bi-directional	Multi-directional
		4	Type	-	Synchronised	-	Asynchronised
		5	Routing	-	Fixed	-	Free
4	Joining	1	Structure	-	Fixed	Changeable - manual	Changeable – auto
		2	Interface	Static/irremovable	Complex/non-standard	-	Simple/standard
		3	Tool magazine	-	None/fixed	-	Changeable
5	Feeding	1	Structure	-	Fixed	-	Modular
		2	Interface	Static/irremovable	Complex/non-standard	-	Simple/standard
		3	Part orientation	None	Passive	-	Active

Table 2. Software convertibility scores.

i	Criteria	0	0.333	0.667	1
1	Openness	Closed	-	Limited	Open
2	Configuration	-	Fixed	-	Modular
3	Auto-adjustment	-	None	-	Available
4	Control type	-	Open-loop	-	Closed-loop
5	Programming	Online	Online – assistive	Offline – vendor specific	Offline – generic



$$C_C = \frac{1}{2}\left(\frac{0.333^2 + 0.333^2 + 1 + 0.667^2 + 1}{5} + \frac{0.667^2 + 1 + 1 + 1 + 0.667^2}{5}\right) = 0.656$$

Fig. 2 Convertibility analysis of 6 axis robot manipulator.

do not fall under any of the described categories are not considered in the model.

Figure 2 illustrates an example of convertibility for a 6 axis robot. The robot consists of elements such as actuators, sensors which are integrated to form the component. Unless the robot is modular, its elements, namely the actuators and sensors cannot be assessed at the hardware and software level. Therefore, the robot is considered as a standalone component and further decomposition is not beneficial for the considered model. It is assumed that the robot is mounted to a station and has a workspace appropriate for application. Since the robot is a motion component, the hardware convertibility factors for motion are considered. The robot structural configuration is fixed and has a non-standard interface with the station. It has variable motion path due to the vast workspace and ability to move to any point in that space. In this example, the robot software is limited in its openness as only certain parameters of the software can be modified. The robot movement is guided by a vision system that enables the robot to adjust according to changes in surroundings. This is captured by the criterion 'auto-adjustment'. In cases of fixed automation which lack flexibility, the score will be 0.333 from the Table 2. The robot, inherently has closed-loop control because of the use of servo motors and the programming is done through offline vendor specific software. Accordingly, the component convertibility of the robot is calculated as 0.656.

3.2. Layout convertibility

Layout convertibility is defined as the ability to change the configuration and/or the part routing to accommodate new product variants. In this study, the system layout is represented as a network, with the nodes representing AWSs and the edges representing part flows. The layout convertibility is defined as the average of the indices describing various aspects of the system layout. Accordingly, it is calculated by Eq. 5.

$$C_L = \frac{L_A + L_C + L_R}{3} \quad (5)$$

3.2.1. Autonomy index, L_A

The layout autonomy index is used to express the system's capability to be autonomous and not be affected or shutdown when conversion in a sub-system takes place. Accordingly, the layout autonomy is high if the system configuration is parallel since there is possibility of re-routing when a sub-system is shut down for conversion. L_A is calculated as Eq. 6.

$$L_A = 1 - \frac{\sum_{k=1}^{N} N_K}{N^2} \quad (6)$$

3.2.2. Connectivity index, L_C

According to [10], the degree of convertibility can be understood by evaluating the routing connections. This approach has been adapted in this research to assess the impact of the routing connections on the layout convertibility. The connectivity index is defined as a function of the existing number of material flow connections and the theoretical maximum and minimum number of flow connections (Eq. 7).

$$L_C = \frac{log_2(N_F - N_{AWS})}{log_2(((N_{AWS} + 2)(N_{AWS} + 1) - 2) - N_{AWS})} \quad (7)$$

It is important to note, since it is impractical to achieve theoretical maximum in real industrial scenarios, a logarithmic function is used to avoid unrealistic scoring for relatively low number of flow paths.

3.2.3. Replication index, L_R

Replication index is adapted from the study proposed by [10]. It is defined as the minimum number of AWSs that have the same operational capability, thereby enabling production of same product. It indicates the number of new product variants that can be introduced to the layout without stopping current production and it is calculated as (Eq. 8), where N_R represents the minimum number of replicated stations.

$$L_R = 1 - \frac{1}{N_R} \quad (8)$$

3.2.4. Illustrative example

An example of three types of layout configurations, each consisting of ten AWSs is depicted to explain the calculation of layout complexity (Fig. 3). Case A represents the stations are arranged in a parallel configuration with an index table transporting the product to all the stations. In case B, two gantries and three index tables are used for material handling. The layout configuration is hybrid with few stations in parallel and few in serial. Case C shows the stations arranged in a serial configuration with product being processed in each station before they can enter the next. Material transport between stations is with a modular conveyor system. Accordingly, the three cases are subjected to the layout convertibility indices and the results are shown in Table 3. It is assumed in this example, that all the part flow directions are unidirectional. From the table, the serial line has poor score for all three indices of layout convertibility. This is because all stations are dependent on one another, significantly reducing its convertibility. This indicates

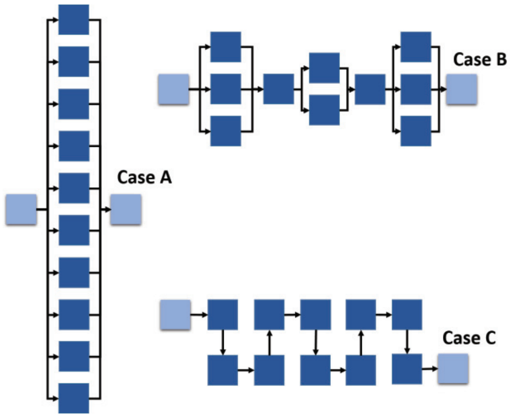


Fig. 3 Sample layout configurations of ten AWSs.

that when a station in serial line is shut down for reconfiguring it, the whole line shuts down.

In comparison, the hybrid configuration scores represent more convertibility than the serial configuration. This is due to the parallel stations that exist in the configuration. The parallel configuration has good scores for all three indices owing to the stations capability of behaving independent of the other stations during operation. Although this indicates the possibility of producing multiple variants in the same system, making it suitable for managing product variety, it is often impractical to be implemented due to the cost involved.

Table 3. Example layout convertibility calculations.

Case	L_A	L_C	L_R	C_L
A	1	0.481	0.9	0.794
B	0.82	0.374	0	0.398
C	0.1	0	0	0.033

4. Use case

The test case demonstrated comprises of nine AWSs and two MHUs, representing eleven subsystems in total for battery module assembly, as shown in Figure 4. Material handling sub-system 1 helps transportation of batteries, busbars, module covers and accessories from the warehouse to the assembly area and vice versa. Material handling sub-system 2 comprises of the modular conveyor unit, that transports the products between the stations. AWSs 1 and 2, perform the same operations of handling batteries and inserting them into the battery trays. AWSs 3 to 9 perform unique operations with each station having a defined operational capability. Stations 3, 4, and 5 locate the top battery tray, insert and tighten nuts, and join sub-modules respectively. Stations 6, 7 and 8 perform busbar locating, pulse arc welding and thermal pad assembly respectively. However, the need to perform busbar assembly and welding on the other side of the module, demands a reorientation operation. Therefore, the module is re-routed to station 5 where the module rotation is performed, after which it passes through the same sequence of assembly operations after which the module cover is assembled in station 9. The case study establishes the convertibility measurement for the conceptualized system and identifies aspects of system that should be considered for re-design. The component convertibility (Eq. 4) is calculated, according to the example shown in Fig 3, for each component present in a subsystem. From Fig. 4, ASW 1 is a subsystem consisting of 4 components and each of them have a convertibility score. This value is later input to Eq. 3 to obtain the convertibility score for each subsystem. Equation 2 is then utilized to find the overall equipment convertibility. In a similar manner, the layout convertibility assessment is performed using Eq. 5, 6 and 7 and can be visualized in Fig. 4. Finally, the system convertibility is evaluated using Eq. 1. It is important to bear in mind that the components should be classified as per section 3.1, and those components that are designed for a specific product, (e.g. work holders, pallet) are to be ignored.

4.1. Results and discussions

The results of the equipment and layout convertibility for the test case is shown in Fig. 4, from which the following can be inferred. MHU 2 and AWS 8 have low convertibility values and MHU1, AWSs 5, 7, and 9 have relatively high convertibility values. The low score of MHU2 is attributed to the conveyor and pallet locator, as can be seen from the component convertibility assessment. On the other hand, the high scores of MHU1 can be attributed to use of an AGV and a 6 axis robot in the system. The autonomy index value is calculated considering the possibility of interchanging AWS 1 and 2, and the connectivity index is calculated bearing in mind that the product can be routed to station 5 from station 8. The layout convertibility measure points out the inability for conveyor direction reversal and high level of station dependency. From Fig. 4, the equipment convertibility score is a bit higher than the layout convertibility score. This is due to the use of 6 axis robots in most of the sub-systems, however the absence of parallel stations and bi-directional product flow reduces the layout convertibility.

5. Conclusion and future work

In this paper, an approach based on heuristics is demonstrated with a battery module assembly test case and it is believed to have the following advantages *i*) ability to quickly assess designs that are detailed, as well as those that lack detail *ii*) reduced effort and cost involved to do the assessment *iii*) the practicality due to the component-based evaluation making it highly suitable for validating initial designs *iv*) quantification of a single design or comparison of multiple designs *v*) supporting optimisation of large assembly systems, where it is tedious to keep track of components used.

The research is an ongoing work, and the subjectivity of evaluating the different hardware and system will be reduced by optimizing the model and calibrating the scoring system for numerous test cases. Although currently, all components are assumed to be equal, empirical study will be done in the future to identify optimum weights for the different components. Additionally, the impact of convertibility at system level on the reconfiguration at the higher level of supervision control for scheduling, production execution etc. form part of future work.

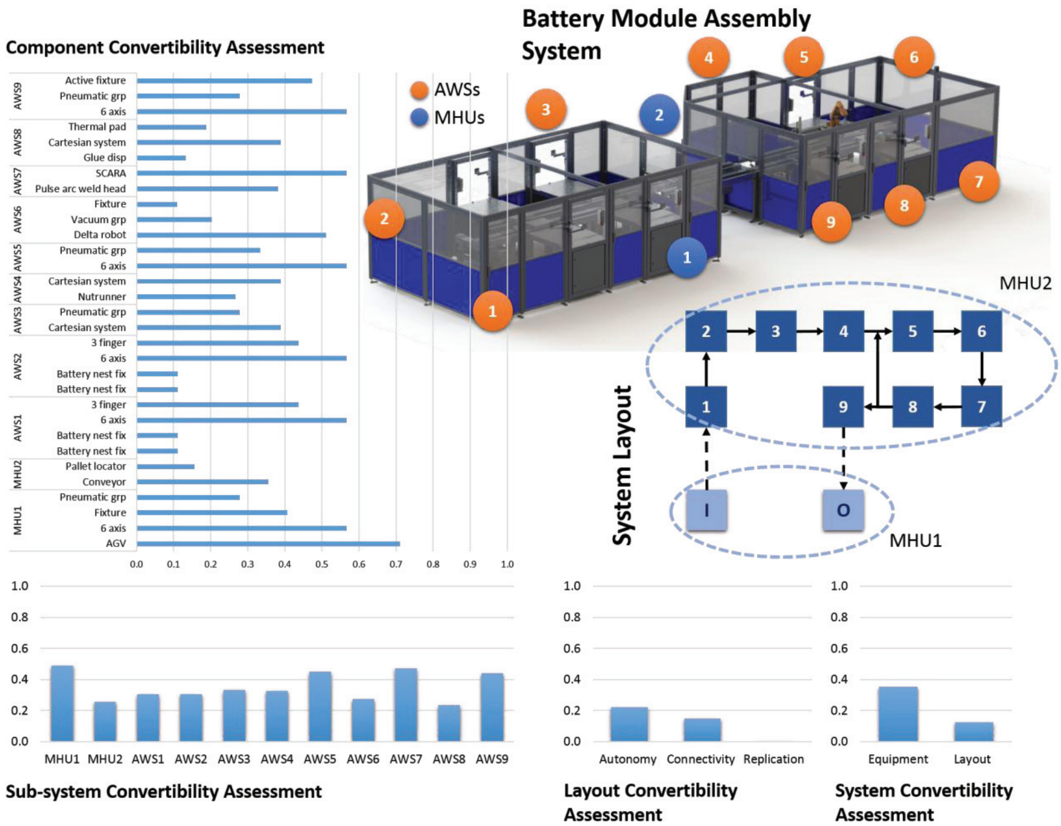


Fig. 4 Illustration of case study ($w_E = 0.5, w_L = 0.5$).

Acknowledgement

The authors are grateful for the support of EPSRC through Knowledge Driven Configurable Manufacturing (KDCM) research project and InnovateUK funded research project AMPLiFII (Automated Module-to-pack Pilot Line for Industrial Innovation). The authors are most thankful to Neal Armstrong and Mussawar Ahmad whose valuable comments helped improve the manuscript.

References

[1] Hu SJ. Evolving paradigms of manufacturing: From mass production to mass customization and personalization. Procedia CIRP 2013;7:3–8.

[2] Putnik G, Sluga A, Elmaraghy H, Teti R, Koren Y, Tolio T, et al. Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap. CIRP Ann. - Manuf. Technol., vol. 62, CIRP; 2013, p. 751–74.

[3] Koren Y, Shpitalni M. Design of reconfigurable manufacturing systems. J Manuf Syst 2011;29:130–41.

[4] Hsu W, Liu B. Conceptual design: issues and challenges. Comput Des 2000;32:849–50.

[5] Wahab MIM, Stoyan SJ. A dynamic approach to measure machine and routing flexibilities of manufacturing systems. Int J Prod Econ 2008;113:895–913.

[6] Chryssolouris G. Flexibility and Its Measurement. CIRP Ann - Manuf Technol 1996;45:581–7.

[7] Farid AM, McFarlane DC. A Tool for Assessing Reconfigurability of Distributed Manufacturing Systems. IFAC Proc Vol 2006;39:523–8.

[8] Hasan F, Jain PK, Kumar D. Machine Reconfigurability Models Using Multi-Attribute Utility Theory and Power Function Approximation. Procedia Eng 2013;64:1354–63.

[9] Farid AM. Measures of reconfigurability and its key characteristics in intelligent manufacturing systems. J Intell Manuf 2014.

[10] Maler-Speredelozzi V, Koren Y, Hu SJ. Convertibility Measures for Manufacturing Systems. CIRP Ann - Manuf Technol 2003;52:367–70.

[11] Lafou M, Mathieu L, Pois S, Alochet M. Convertibility indicator for manual mixed-model assembly lines. Procedia CIRP 2014;17:314–9.

[12] Goyal KK, Jain PKK, Jain M. Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS. Int J Prod Res 2012;50:4175–91.

[13] Youssef AMA, ElMaraghy HA. Assessment of manufacturing systems reconfiguration smoothness. Int J Adv Manuf Technol 2006;30:174–93.

[14] Ahmad M, Alkan B, Ahmad B, Vera D, Meredith J, Bindel A, et al. Hydrogen fuel cell pick and place assembly systems : Heuristic evaluation of reconfigurability and suitability. CIRP Ann - Manuf Syst 2016;0:25–7.

[15] Lohse N. Towards an ontology framework for the integrated design of modular assembly systems. 2006.

[16] Alkan B, Vera D, Ahmad M, Ahmad B, Harrison R. Design evaluation of automated manufacturing processes based on complexity of control logic. 26th CIRP Des Conf 2016;50:141–6.

[17] ElMaraghy H a. A Complexity Code for Manufacturing Systems. ASME 2006 Int. Manuf. Sci. Eng. Conf., vol. 2006, 2006, p. 625–34.

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 63 (2017) 604 – 609

www.elsevier.com/locate/procedia

The 50th CIRP Conference on Manufacturing Systems

An application of physical flexibility and software reconfigurability for the automation of battery module assembly

Malarvizhi Kaniappan Chinnathai[a]*, Till Günther[b], Mussawar Ahmad[a], Cosima Stocker[b], Lukas Richter[b], David Schreiner[b], Daniel Vera[a], Gunther Reinhart[b], Robert Harrison[a]

[a]Automation Systems Group, WMG, University of Warwick, CV4 7AL, West Midlands, UK
[b]Institute for Machine Tools and Industrial Management (iwb), Technical University of Munich (TUM), Boltzmannstr. 15, D-85748 Garching, Germany

* Corresponding author. *E-mail address: M.Kaniappan-Chinnathai.1@warwick.ac.uk*

Abstract

Batteries are a strategic technology to decarbonize conventional automotive powertrains and enable energy policy turnaround from fossil fuels to renewable energy. The demand for battery packs is rising, but they remain unable to compete with conventional technologies, primarily due to higher costs. Major sources of cost remain in manufacturing and assembly. These costs can be attributed to a need for high product quality, material handling complexity, uncertain and fluctuating production volumes, and an unpredictable breadth of product variants. This research paper applies the paradigms of flexibility from a mechanical engineering perspective, and reconfigurability from a software perspective to form a holistic, integrated manufacturing solution to better realize product variants. This allows manufacturers to de-risk investment as there is increased confidence that a facility can meet new requirements with reduced effort, and also shows how part of the vision of Industry 4.0 associated with the integration and exploitation of data can be fulfilled. A functional decomposition of battery packs is used to develop a foundational understanding of how changes in customer requirements can result in physical product changes. A Product, Process, and Resource (PPR) methodology is employed to link physical product characteristics to physical and logical characteristics of resources. This mapping is leveraged to enable the design of a gripper with focused flexibility by the Institute for Machine Tools and Industrial Management (iwb) at the Technical University of Munich, as it is acknowledged that mechanical changes are challenging to realize within industrial manufacturing facilities. Reconfigurability is realised through exploitation of data integration across the PPR domains, through the extension of the capabilities of a non-commercial virtual engineering toolset developed by the Automation Systems Group at the University of Warwick. The work shows an “end-to-end” approach that practically demonstrates the application of the flexibility and reconfigurability paradigms within an industrial engineering context.

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems

Keywords: battery module assembly; reconfigurability; flexibility; virtual engineering

1. Introduction

Efforts are being made to transition society towards renewable energy technologies, driven by policy and legislation, due to the threat posed by increases in greenhouse gas emissions and combustion pollutants [1]. It is estimated that currently 25% of CO_2 emissions can be attributed to the transport sector; this is projected to rise to 50% by 2030 if current trends continue [2]. Electric vehicles are a potential solution as sufficient deployment will reduce pollutants, greenhouse gases, and offer significant well-to-wheel efficiency improvements [3]. There are a range of automotive propulsion system configurations ranging from mild-hybrids to purely electric systems. Irrespective of architecture however, batteries remain a common key enabler of electrification for energy storage within and external to the automotive sector [4]. A breadth of applications for battery technologies is anticipated within the coming years which bring with them a broad range of potential variants and product types that may need to be produced by a single production system. The degree of variety is difficult to predict and so engineers are compelled to design manufacturing systems to be able to accommodate change. This need aligns with the vision of Industry 4.0, where connectivity across all levels of the business and through the

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
doi:10.1016/j.procir.2017.03.128

product and system lifecycles facilitates manufacturing agility and proactivity [5].

Two major phases of a system lifecycle are design and re-engineering/reconfiguration. At the initial design phase, a number of considerations need to be made, one of which is to try and anticipate the breadth of capability the system needs with respect to product requirements. Reconfiguration phases are often driven by changes to the product or new product introduction. In order to reduce the time and accompanying costs associated with this phase, it is beneficial to know i) the nature of the system changes, and ii) a mechanism for executing the change with minimal human intervention. Some common existing paradigms associated with change within manufacturing systems are flexibility and reconfigurability. However, formal implementation of these concepts within the engineering workflow during the system design and reconfiguration phases is limited. In line with the vision of Industry 4.0, this study proposes that the integration of product realisation domains (Product, Process and Resource (PPR)) through lifecycles within engineering tools is fundamental in managing change. The approach is demonstrated on the introduction of a new variant in a battery module assembly system.

2. Literature Review

2.1. *Digital Manufacturing*

Digital Manufacturing is one of the disciplines within Product Lifecycle Management (PLM) [6], where Computer Aided Design (CAD) and Computer Aided Engineering data plays a vital role in managing products and systems through their respective lifecycles. The concept of Digital Planning Validation is discussed in [7], where the validation of a product's produce-ability is done parallel to the production planning phase in a digital environment. Having validated the plans virtually, training materials for operators can be generated and used. Digital Mock-Ups discussed in [8] are used to simulate a production system to verify and validate system configurations, layouts, and process plans. Integration of digital models with the physical system is done during the commissioning phase, often to validate programmable logic controller (PLC) software. This has been demonstrated in [9] through the use of Logic Control Modeling connected to DELMIA Automation V5, and Tecnomatix eM-PLC from Siemens. Beyond this point, however, digital models see limited use as they are not maintained post the build and commissioning phases. Thus, during reconfiguration there is limited support from digital manufacturing or PLM tools. For example, translation of changes in product features through to machine control parameters within PLC programs remains an entirely manual process, supported through ad-hoc methods [10,11]. As a result, despite the benefits of the digital manufacturing paradigm at the design phase, its value with respect to supporting and executing flexibility and reconfigurability on the shop floor is limited.

2.2. Flexibility and Reconfigurability

There are many definitions for flexibility, reconfigurability, and related terms within the literature. Following ElMaraghy, for example, the ability of production systems to be adaptable to continuous changes is described as changeability [12]. Forming a subcategory of changeability, flexibility is related to the assembly system, while reconfigurability refers to the entire production area including logistics [12]. The authors have chosen the definiton put forward by Koren ([13,14]): "flexibility is the general ability to respond to changes in production volume or product variants in a fast and global cost efficient way without changing elements of the production line" [13], as it aligns with the approach presented in this paper. A design framework for flexible systems is proposed in [24]. It consists of four stages supported by process management. The baseline design assists designers in the early design process using known configurations. This is followed by the uncertainty recognition which is to help identify the range of flexibility. In the concept generation phase, concepts are generated to handle the identified range of flexibility. Finally, designers analyse and evaluate the generated concepts. The proposed taxonomy and further literature [25] focus on the system level. A detailed methodology for the design of flexible system components for a production system is absent in the literature.

Design methodologies for flexible production system are needed to achieve reconfigurability. Reconfigurability is considered a subset of flexibility [15]. It is the ability to change the capability of production equipment by adding or removing functional elements in a short time and with low effort to meet new requirements within a part family [13]. Reconfigurability within the software domain is addressed by [16] who discusses issues faced with automatic software reconfiguration such as: the absence of a formal procedure for implementation, limited application of the available methods, and the need to reconfigure all processes simultaneously. According to [17], within the context of manufacturing, software reconfiguration for control systems is considered a key enabler for reconfigurable manufacturing systems (RMS). Self-adapting control software is created through integration with a mechatronic model, reducing post reconfiguration system ramp up time [17]. A reconfigurable control architecture that can adapt to changes has been proposed by [18], in which component based development has been combined with holonic manufacturing system to provide an architecture for a decentralized manufacturing system. In [19], a framework is proposed to translate the assembly sequence change necessitated as a consequence of product variant introduction to the control system logic through virtual engineering tools. In [20], a PPR ontology knowledge-driven approach, enables increased reactivity to change. Despite the advancements in software reconfiguration, according to [21], the inability of the current PLCs to help realise RMS, is an inhibitor to the implementation of control software reconfiguration. One reason for this is the current use of the IEC 61131-3 standard as it does not favour dynamic reconfiguration. However, the IEC 61499 standard is sought to address this issue as it more suitable for reconfiguration [22], however gaining industrial acceptance for this standard has proved to be a challenge [23]. Despite these advances, reconfiguration at the field device level still needs to be supported by the wider engineering lifecycle, which at present lacks suitable engineering tools and methods [17].

2.3. Summary

The importance of flexibility and reconfigurability is recognized, but due to limited formal, structured engineering processes and links across domains, true realisation of these paradigms remain hamstrung by inefficient workflows. Therefore, this paper proposes a PPR framework that demonstrates i) how manufacturing system components should be designed to have sufficient flexibility for the anticipated product variety i.e. focused flexibility, and ii) an engineering workflow that supports reconfiguration through the use of component-based virtual engineering tools.

3. Approach

3.1. PPR framework

A **PPR framework** is used in this work as described in Fig. 1. At the highest level, the product drives the process, which in turn drives the resource. At the point of resource existence in the physical (or digital) world, it begins to constrain the process which in turn constrains product design. This set of assumptions is used to drive the **component design process** with sufficient flexibility to accommodate a range of product variants and consequently, a range of process parameters through a requirements list (Section 3.2). The design information is instantiated into a set of **virtual engineering** tools which support the system through its lifecycle. As such, common data models can be used both in the design phase and later in the operation phase to support reconfiguration, exploiting the flexibility designed into the system (Section 3.3).

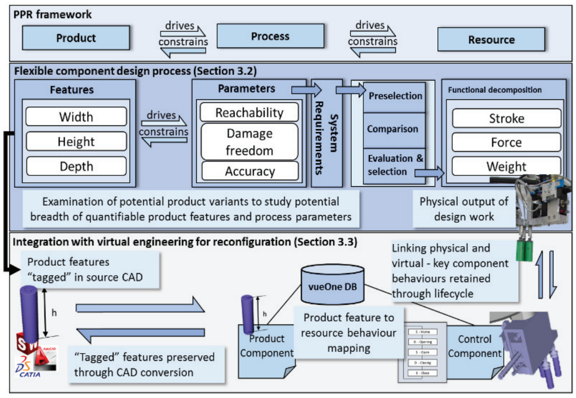


Fig. 1. PPR framework with flexible manufacturing system component design, and reconfigurability through virtual engineering.

3.2. Product/Process parameter selection for machine component design

A requirements list based on product/process parameters is created and developed iteratively. Firstly, general requirements e.g. safety, environment, interfaces etc., are identified; this is a system level view. Next, a deep-dive on product requirements is carried out, analysing all members of a focused product family. At this point, key product **features** are extracted from the overall parameter set e.g. width, height, depth (Fig. 1), to extract basic product designs in the form of topologies. These topologies build the basis for a heuristic solution search.

Next, the process **parameters** are investigated which include: reachability, freedom of damage, and positional accuracy (Fig. 1). After a general **preselection**, the derivation of the requirements is classified into demand and request by the comparison of couples (**comparison**, Fig. 1). A Pareto analysis is conducted to split mandatory from optional requirements to reduce complexity. Once all appropriate requirements have been captured, Resource domain parameters are defined. The physical description of necessary skills is derived from range definitions. The necessary skills identified define the functional structure of the Resource component. Through **functional decomposition** into subfunctions, operating principle selection is enabled using a morphological analysis. Based on the set of operating principles, potential concepts are generated. Any concept to be further detailed is selected through a utility analysis which uses the evaluation criteria from the initial requirements list. During the selection process, those solutions that offer the ability to rapidly reconfigure through software i.e. mechatronics, are most favourable, despite not having lowest initial investment cost. System reconfiguration offered through software modifications provides compatibility with the Industry 4.0 vision. The following section describes how engineering tools can use design data to support reconfiguration to exploit the flexibility designed into the system.

3.3. vueOne toolset for supporting reconfiguration

vueOne is an engineering toolset that supports the lifecycle of a production system. It was developed by the Automation Systems Group at the University of Warwick. Within the tools, extensible component-based data models support process planning, system configuration, code generation and deployment, commissioning, maintenance, operational analytics, and system reconfiguration [26]. Geometries for system components are converted from native CAD formats to VRML/X3D and form a part of a software component within the tool, uniquely identified through an ID. This assists the identification and management of the components in later stages of the product lifecycle. During the process planning phase, system behaviour is modelled through the combination of kinematics and state transition diagrams (STDs) that are IEC 61131 compliant. Using a mapper module within the tools, these behavioural models are **mapped** to function blocks for the automatic generation of programmable logic controller (PLC) code and virtual commissioning through OPC-UA client connectivity. A specific type of software component within the tools created for this work is the "**Product Component**" which contains the product geometry and the key product feature information described in 3.2 (Fig. 1). Although product geometry could previously be imported in the tools, there was no mechanism for enriching the information i.e. key product features/characteristics identified by the design phase. These key product features are mapped to parameters of machine component states, i.e. actuators, by the user. This link is preserved within the database of the engineering tools (**vueOne DB**, Fig. 1). Once this link exists, it is maintained as each respective component has a constant ID through its lifecycle. Thus, if a given product design changes, the machine behaviour is also modified due to the explicit **link** between data models at a fine level of granularity. Of course, it is necessary for the native product CAD format to originally have this feature "**tagged**" in a way that prevents loss during conversion (Fig. 1). At present, this issue has not been fully resolved but it is expected that the Product and Manufacturing Information (PMI) which is supported by several CAD formats would be key. The formal, explicit link between the respective PPR domains through virtual engineering tools presents the ability to i) identify whether the product features of a new variant fit into the system range through rules, ii) identify the impact of product attribute change on the resource domain through visualisation and system behaviour simulation, and iii) modification of PLC software with the confidence that it will meet requirements from the product – resource coupling. In this research, items ii) and iii) are tested in the case study.

4. Case Study

4.1. Experimental setup

The framework and approach described in Chapter 3 is applied to the battery module assembly station at the Technical University of Munich (TUM), pictured in Fig. 2. The battery cells are handled by a collaborative robot (1) mounted on a linear axis (2) in order to increase the robot range. The feeding line (3) houses battery module components. The battery modules are assembled on a central mounting station (4). The robot is equipped with a flexible cell gripper designed using the method described in Section 3.2. The application of the methodology is explained in Section 4.2.

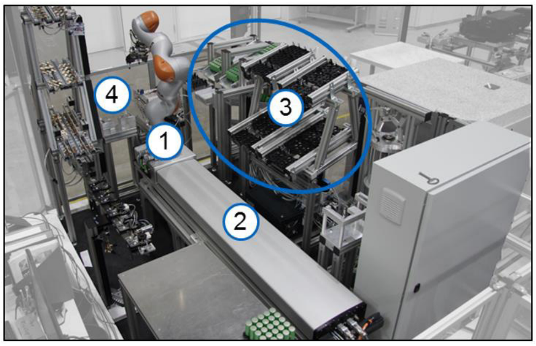


Fig. 2. Flexible and modular assembly station for battery modules.

The global requirement for the assembly station is to accommodate the assembly of battery modules for stationary energy storage and automotive applications. The different module use cases have different sets of design requirements. All components in the cell have been developed to suit a broad range of possible battery modules. In this case study, two different modules are to be assembled successively. The stationary energy storage module, product 1, consists of six cylindrical lithium-ion cells type 26650, which are arranged in a triangular configuration on a cell holder. For heat management purposes, there is a gap between the cells for air-cooling. The battery modules for the automotive industry, product 2, consist of six prismatic lithium-ion cells type PHEV1 which were developed at the TUM in the project ProLIZ. Liquid cooling of cells necessitates direct contact between the prismatic cells. The following case study demonstrates the application of the flexible component design methodology and how the introduction of product 2 is accommodated by the gripper from a mechanical flexibility and software reconfigurability perspective.

4.2. Application of component design method to the gripper

Grippers can be categorized into three flexibility domains by [27]: i) adaption to geometry and/or mass of work pieces, ii) change of functional elements, and iii) self-adaption to object-specific characteristics. Flexibility can be achieved with universal grippers that can adapt to every gripping operation and special grippers. The complexity of a gripper increases with the rise of mechanical flexibility [28], therefore its physical implementation has to be reduced and enhanced otherwise. The design methodology for flexible manufacturing system components is applied to the gripper for the system described in 4.1.

First, the general requirements list is created which focuses on avoiding cell damage and applying constant force. The product family within the context of battery modules is examined through a review of all possible cell types present in the market. Multiple criteria are researched, e.g. characteristic width of 120-173 mm for prismatic cells, 70-150 mm for pouch cells and 18-26 mm diameter for cylindrical cells. Having determined the ranges, specific process requirements are extracted, primarily oriented towards the mounting direction depending on the cell type. Cylindrical cells require uniaxial vertical mounting, while prismatic and pouch cells demand multiaxial mounting techniques. The requirements are divided into mandatory and optional criteria. Based on the requirements list, the functional decomposition is executed leading to the identification of functions such as gripper adaption to different cell geometries. Operating principles for each function were collected, for this use case, multipoint jaws and adjustable vacuum cups are selected. Two concepts were designed based on the aforementioned operating principles.

Both concepts were evaluated using a utility analysis based on the requirements list. The gripper equipped with multipoint jaws was excluded from the mechanical construction because of its inability to grip pouch cells in the sealed area, which is needed for specific handling situations. Applying the Product-Process mapping on the mechanical design of the gripper, three vacuum cups were selected enabling the handling of three round cells simultaneously, enhancing process efficiency. Moreover, the handling of pouch and prismatic hard case cells was ensured due to the extended gripping surface.

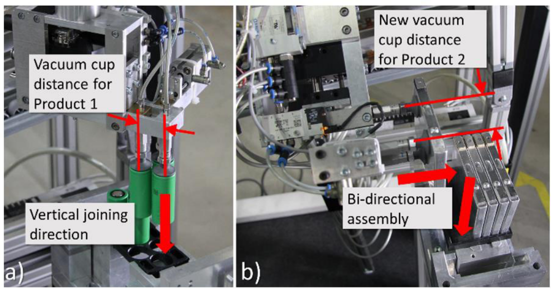


Fig. 3.(a) Gripper behaviour for product 1 and (b) new gripper behaviour achieved through software reconfiguration via engineering tool integration.

The final design consists of a fixed vacuum cup, a vacuum cup on a pneumatically driven linear axis, and a vacuum cup on a programmable electrically driven linear axis. The electrical axis contains a JUNG QuickPos® linear motor, actuated by a FAULHABER motion controller. A serial RS232-interface is used to communicate the target value to the linear motor. To ensure the handling of cells within the identified dimension range, the distance between the cups can be varied between 21.5 mm and 71.5 mm.

4.3. Mechanical flexibility

Due to the three replaceable vacuum cups, the gripper possesses adequate mechanical flexibility for the product family. Handling of cylindrical batteries is achieved through gripping centrally at the top with a distance of 29.5 mm between the cups, whereby three cells can be processed simultaneously (Fig. 3a). Cells for product 1 are picked and placed with a vertical motion. The prismatic cells of product 2 are gripped at the face with the largest surface area. The three vacuum suckers are reoriented at equal distances from the center of mass of the cell, resulting in a distance of 61.5 mm between the cups. Due to the different cooling principle of product 2, the production process also has to be changed: the vertical joining is transferred to a bi-directional joining, composed of a vertical movement, followed by horizontally joining the cells to achieve contact between them (Fig. 3b). Note that the bi-directional nature of the process is largely handled by the robot, the handling process itself is enabled by the gripper's flexible design. The design method has synthesized a broad spectrum of product and process features/characteristics into a single efficient design. The software reconfiguration necessary for the introduction of the new product is described in the following section.

4.4. Software reconfiguration

The initial conditions of the virtual model in the engineering tools are aligned to those sets of behaviours matching the requirements of product 1, e.g. the spacing between the vacuum cups of the gripper. When the production is now changed from a battery module of type 1 to type 2, new code needs to be uploaded to the PLC . Therefore a reconfiguration of the software is required due to the different requirements of product 2 compared to product 1: the vaccum cups need to change their positions. Figure 4 illustrates how data is taken from the **source CAD** file, pulled into state behaviour of system components and control code for the **PLC** is **generated** and **deployed** for product 2.

It is envisioned that the product designer would be informed which features to annotate or tag based on a set of rules created as an output of the system component design phase decribed in 4.2. The source CAD file is converted to VRML/X3D through a convertor in the engineering tools. The **annotation** is then present in the file (typically VRML/X3D does not have support for annotations, but within the toolset this is overcome through explicit insertion). When the user creates the Product Component within the vueOne toolset, the tool parses the **VRML**/X3D file for "tagged" features which then formally form part of the **Product Component** data model. Once the product feature information is within the Product Component data model, it is accessible by the **STD** of any controllable component i.e. actuator, in the engineering tool (**vueOne DB**, Fig. 4).

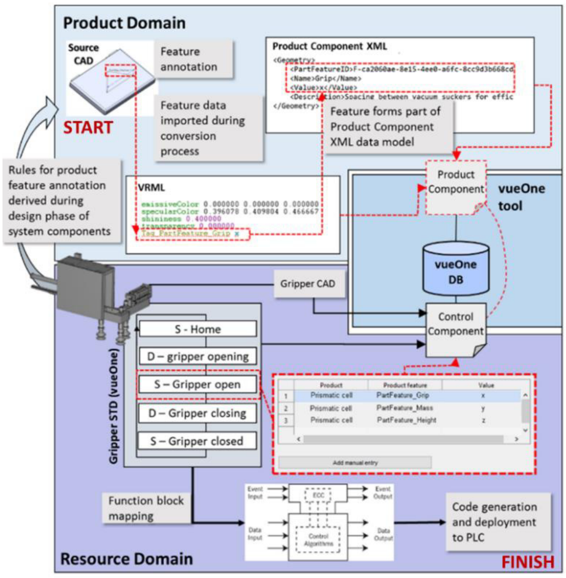


Fig. 4 Workflow for capturing product feature information and mapping to gripper behaviour. Red dashed lines indicate new workflow developed through this work, while black lines correspond to existing tool capability.

When the user imports the Product Component data model for product 2 into the virtual system, the mappings between product 1 and the STD are replaced. The user must then navigate to the gripper state associated with gripping and access product features of product 2. "PartFeature_Grip" is selected which has a value of 61.5mm. Now, an explicit link has been formed between the state of the gripper and the product feature. If the feature is changed in the VRML/X3D, the machine behaviour changes as well. This explicit mapping facilitates more rapid product and process validation, as well as system reconfiguration.

4.5. Evaluation

The case study has demonstrated how the integration between the PPR domains supports the design and reconfiguration phases of an assembly system. The approach in this study has successfully demonstrated that the gripper has sufficient flexibility to handle both cylindrical and prismatic cells with small modifications to the software. Using the methodology, the complexity of the gripper's design has been limited while still providing the necessary degree of flexibility. However, the analysis was focused on gripper design, and therefore a predefined perspective was imposed. Alternative processes may require a different set of product/process parameters to be considered. This could result in an extensive approach to system design to ensure sufficient flexibility.

Classically, modifying the behaviour of drives in an industrial application would be done on the human machine interface or through a new program on the PLC, and there would be either a very limited or no link to product data. The vision of Industry 4.0 is, in part, one of data integration. In this work, this has been achieved through the use of virtual engineering tools which integrate i) the physical world with the associated digital model, and ii) key product characteristics with machine component behaviour. The former further demonstrates the importance of virtual engineering, while the latter forms a key contribution of this work. However, some manual steps still remain. Although many CAD formats support PMI i.e. ISO 10303 STEP, ISO 14306:2012 JT, standards associated with how such information should be described do not extend into the domain of product assembly. For example, ASME Y14.41 focuses on the presentation of geometrical dimensioning and tolerancing data. Standards associated with defining assembly processes i.e. VDI 2860, are typically not present within CAD software. This results in inconsistent descriptions of tagged features and thus conventional conversion software would be unable to identify key information. This problem could potentially be overcome through the use of Semantic Web Technologies, where meaning concerning the nature of a tagged feature is preserved. Alternatively, integration between CAD tools and vueOne could be achieved through a software interface that writes PMI data directly to the database.

5. Conclusion and Further Work

The aim of this work was to demonstrate how challenges associated with reduced product lifecycles and increasing product variety, particularly within the context of batteries, could be overcome. The authors proposed a PPR framework which considered potential product variants to instill mechanical flexibility into manufacturing system components. On creation of the physical system, future product design environments would have rules which supported the tagging of appropriate product data. Virtual engineering tools then integrate digital product data to digital representations of the physical system. This facilitates pre-validated software reconfiguration realising increased manufacturing responsiveness with reduced risk. The framework has been expanded to an approach that has successfully demonstrated new product introduction on an assembly system. This work demonstrates a mechanism to achieve this through the design and (re)engineering lifecycles of products and systems. Future work includes improved integration between source CAD and virtual engineering tools for manufacturing systems, and further validation of the method associated with design of flexible system components.

Acknowledgements

The authors gratefully acknowledge the support of the Bavarian Ministry of Economic Affairs and Media, Energy and Technology through the “EEBatt” project, Innovate UK through the AMPLiFII project (ref: 102490), and UK EPSRC through KDCM (ref: EP/K018191/1) and iCASE PhD studentship (ref: 101980) for the funding of our research.

References

[1] Thiel C, Perujo A, Mercier A (2010) Cost and CO2 aspects of future vehicle options in Europe under new energy policy scenarios. *Energy Policy* 38(11):7142–51.

[2] (2009) *Transport, energy and CO2: Moving toward sustainability*. OECD IEA, Paris.

[3] Barkenbus J (2009) Our electric automotive future: CO2 savings through a disruptive technology. *Policy and Society* 27(4):399–410.

[4] Garcia-Valle R, (Ed.) (2013) *Electric vehicle integration into modern power networks*. Springer, New York.

[5] Berry WL, Cooper MC (1999) Manufacturing flexibility: Methods for measuring the impact of product variety on performance in process industries. *Journal of Operations Management* 17(2):163–78.

[6] Chryssolouris G, Mavrikios D, Papakostas N, Mourtzis D, Michalos G, Georgoulias K (2009) Digital manufacturing: History, perspectives, and outlook. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 223(5):451–62.

[7] Wöhlke G, Schiller E Digital planning validation in automotive industry. *Proceedings of the IFIP WG 5.7 Working Conference on Human Aspects in Production Management*.

[8] Maropoulos PG, Ceglarek D (2010) Design verification and validation in product lifecycle. *CIRP Annals - Manufacturing Technology* 59(2):740–59.

[9] Park CM, Bajimaya SM, Park SC, Wang GN, Kwak JG, (Eds.) (2006) *Development of Virtual Simulator for Visual Validation of PLC Program*. IEEE, Piscataway, NJ.

[10] Baqar Raza M, Harrison R (2011) Design, Development & Implementation Of Ontological Knowledge Based System For Automotive Assembly Lines. *IJDKP* 1(5):21–40.

[11] Conti JP (2008) Getting connected. *Engineering & Technology* 3(12):68–71.

[12] ElMaraghy HA, (Ed.) (2009) *Changeable and Reconfigurable Manufacturing Systems*. Springer London, London.

[13] Koren Y (2010) *The global manufacturing revolution: Product-process-business integration and reconfigurable systems*. Wiley, Hoboken, N.J.

[14] Koren Y, Shpitalni M (2010) Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems* 29(4):130–41.

[15] Wiendahl H-P, ElMaraghy HA, Nyhuis P, Zäh MF, Wiendahl H-H, Duffie N, Brieke M (2007) Changeable Manufacturing - Classification, Design and Operation. *CIRP Annals - Manufacturing Technology* 56(2):783–809.

[16] Szepesi R, Ciocârle H An Overview on Software Reconfiguration. *Theory and Applications of Mathematics & Computer Science* 2011(1):74–9.

[17] Dashchenko AI, (Ed.) (2007) *Reconfigurable Manufacturing Systems and Transformable Factories: 21st Century Technologies*. Springer, Dordrecht.

[18] Chirn J-L, McFarlane DC (2000) A holonic component-based approach to reconfigurable manufacturing control architecture. in Tjoa AM, (Ed.). *Proceedings / 11th International Workshop on Database and Expert Systems Applications: 4 - 8 September, 2000, Greenwich, London, United Kingdom ; [workshop papers]*. IEEE Computer Soc. Los Alamitos, Calif., pp. 219–223.

[19] Ahmad M, Ahmad B, Harrison R, Alkan B, Vera D, Meredith J, Bindel A (2016) A Framework for Automatically Realizing Assembly Sequence Changes in a Virtual Manufacturing Environment. *Procedia CIRP* 50:129–34.

[20] Ferrer BR, Ahmad B, Lobov A, Vera DA, Lastra JLM, Harrison R (2015) An approach for knowledge-driven product, process and resource mappings for assembly automation. in Jia Q-S, (Ed.). *2015 IEEE International Conference on Automation Science and Engineering (CASE): 24 - 28 Aug. 2015, Gothenburg, Sweden*. IEEE. Piscataway, NJ, pp. 1104–1109.

[21] Mehrabi MG, Ulsoy A, Koren Y (2000) Reconfigurable manufacturing systems and their enabling technologies. *IJMTM* 1(1):114.

[22] Zoitl A, Sunder C, Terzic I (2006) Dynamic reconfiguration of distributed control applications with reconfiguration services based on IEC 61499. in Mařik V, (Ed.). *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and its Applications: DIS 2006 ; 15 - 16 June 2006, Prague, Czech Republic*. IEEE. Piscataway, NJ, pp. 109–114.

[23] Thramboulidis K (2013) IEC 61499 vs. 61131: A Comparison Based on Misperceptions. *JSEA* 06(08):405–15.

[24] Cardin M-A (2014) Enabling Flexibility in Engineering Systems: A Taxonomy of Procedures and a Design Framework. *J. Mech. Des.* 136(1):11005.

[25] Tolio T (2009) *Design of Flexible Production Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[26] Harrison R, Vera D, Ahmad B (2016) Engineering Methods and Tools for Cyber–Physical Automation Systems. *Proc. IEEE* 104(5):973–85.

[27] Heisel U, Willmer R (1992) *Multifunktionale Greifwerkzeuge für die flexible Montage*. Universität Stuttgart.

[28] Bartenschlager J, Hebel H, Schmidt G (1998) *Handhabungstechnik mit Robotertechnik: Funktion, Arbeitsweise, Programmierung*. Vieweg+Teubner Verlag, Wiesbaden, s.l.

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 72 (2018) 796–801

www.elsevier.com/locate/procedia

51st CIRP Conference on Manufacturing Systems

Pilot To Full-Scale Production: A Battery Module Assembly Case Study

Malarvizhi Kaniappan Chinnathai*[a], Bugra Alkan[a], Daniel Vera[a], Robert Harrison[a]

[a] Warwick Manufacturing Group (WMG), University of Warwick, Coventry, West Midlands, CV4 7AL, UK

* Corresponding author. Tel.: +44-7831001684; *E-mail address:* m.kaniappan-chinnathai.1@warwick.ac.uk

Abstract

Electric vehicles are currently on the rise due to environmental and legal concerns. Furthermore, improvements made in battery assembly steadily boosts the efficiency of electric vehicles. A well-prevalent method to overcome the uncertainties that emerge from the ever-changing battery technology, is to assemble products using pilot production lines. However, literature pertaining to the scale-up of pilot production lines for full scale production is scarce. Therefore, in this paper, potential scale-up scenarios for battery module assembly line are proposed in a discrete event simulation software and results are compared. Furthermore, the benefits of the proposed method are discussed with a test case.

Peer-review under responsibility of the scientific committee of the 51st CIRP Conference on Manufacturing Systems.

Keywords: Cyber-Physical Systems; Battery module assembly; Pilot production.

1. Introduction

An important strategy adopted to ameliorate the undesirable effects of greenhouse gases and CO_2 emissions is power train electrification [1]. It is therefore predicted that the demand for electric mobility will slowly rise [2]. Consequently, it is essential for automobile industries to develop competencies in battery technology to remain competitive in the market. A well-prevalent strategy to fulfill this vision, is to build pilot lines to capture knowledge to be transferred for full-scale battery assembly [3]. A key aspect of battery manufacturing and assembly is that, it is currently facing multi-faceted problems arising from high manufacturing cost, unpredictable market, rapid changes in technology, increased number of variants and missing standardization of battery design [4].

Therefore, to overcome these challenges, various studies are being performed at WMG, as part of a suite of on-going research projects to capture knowledge from pilot production lines to support the early validation and verification capabilities for full-scale production, such that process optimisation and best-practice procedures for battery assembly can be quickly established.

With the advent of Industry 4.0, computer simulation is now an established way of improving the lifecycle management of the products by supporting decision-making, scheduling and cost analysis. Discrete-Event Simulation (DES), in particular, has been adopted to perform layout design, analyse operational performance [5] and has established its presence in the manufacturing domain [6]. In the context of battery module assembly, it is essential to simulate the product variants and its effect on material flow; discrete-event simulation can be used for this purpose [7]. Owing to the lack of implementation of such models in battery production, this paper discusses a case of battery module assembly, with the possible scenarios of scale-up for a mixed model assembly line. In this regard, the scale-up policies are integrated with two standard dispatching rules and the resulting scenarios are modelled using a DES software. Relevant statistical methods are used for comparison of the scenarios and the methodology is validated using a test case of two battery module variants. The impact of the product variety and system configuration on the pilot line and its potential scale-up scenarios and the support provided by Cyber-Physical Systems (CPS) in decision making are discussed.

2. Literature Review

In this section the research gap is highlighted by reviewing the available literature in three major areas namely: scale-up principles, battery module assembly and DES modelling. The research trend across these streams are discussed and summarized.

Manufacturing industries face several challenges during the transition of ideas and design from concept development to full-scale production. During this shift, unfavourable disturbances and challenges, such as the i) the inability to increase functionality of stations due to certain constraints ii) lack of knowledge regarding potential material flow issues iii) effect of scale-up on the labour and material feeding etc., can impact the performance of the system. Therefore, it is desirable to detect and prevent these disturbances as early as possible;

2212-8271 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 51st CIRP Conference on Manufacturing Systems.
10.1016/j.procir.2018.03.194

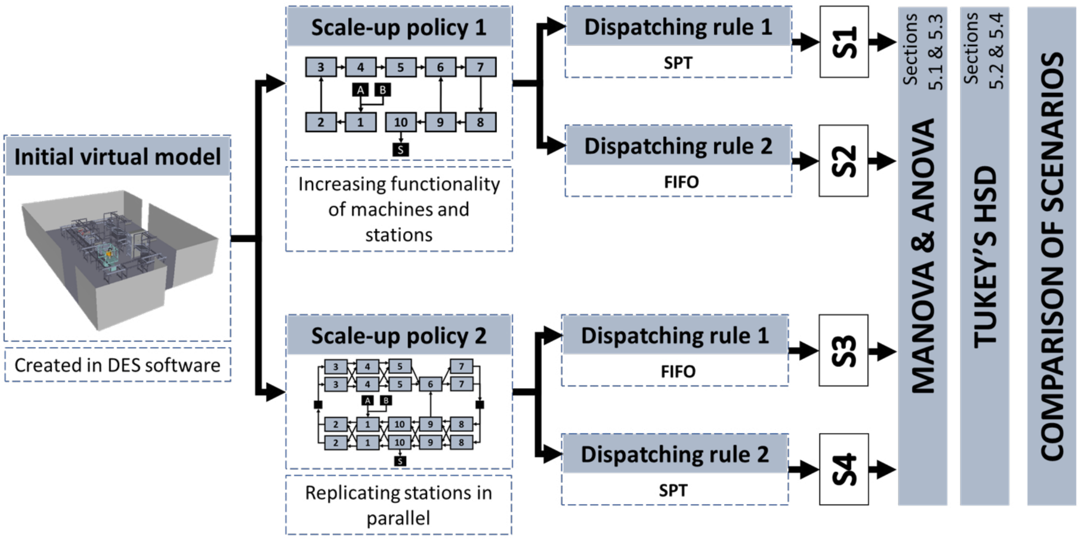


Fig. 1. The proposed methodology.

pilot production lines, which are considered as a training bed for full-scale production can be used for this purpose and, [8] in their research, highlight this issue. The transition from pilot to full-scale production, is not without challenges, hence, it is necessary to adopt strategies to enable and realise this transition. In this regard, [9] discuss two main principles for the implementation of scale-up in a manufacturing system. Moreover, [10], in their research, present a detailed account of the significant aspects and challenges faced in the scale-up of processes in the pharmaceutical industry. Scalability is considered as an important characteristic of Reconfigurable Manufacturing System (RMS); [11] consider an approach for the capacity scaling of RMS supported by optimization techniques to predict the time and extent of scaling necessary. The type of demand scenario that is considered can impact the strategy adopted for capacity scaling and this is discussed by [12]. [13] introduce a methodology to scale system capacity by reconfiguration of the system. Conclusively, studies and research works pertaining to provision of methodology or systematic approach to guide the process of scaling a pilot line are limited.

In the domain of battery assembly, notable research include modelling fault-tolerant control of the system [14], framework for automating the design process in the absence of standards for the battery components [15] and supporting decisions on assembly system design, equipment selection and task allocation [16].

Discrete-event simulation has seen its application in expediting the decision-making process in early production phase by utilizing pre-defined modules in power train electrification scenario [17]. [7] applies the concept of multi-scale simulation in task allocation, buffer size analysis and other operational elements in a battery module assembly case. According to [7], the concepts of simulation have been applied to battery electrode, cell and system modelling, however the realization of simulation in the domain of battery production process has not been well established.

2.1. Summary

Several studies have been conducted regarding the scalability of production lines under different demand scenarios. However, the concept of scaling up a pilot production line has not been widely researched in the context of manufacturing systems. Moreover, there is lack of a formal methodology for realizing a smooth transition from the pilot line to full-scale production. Simulation and modelling have established digitization of design data and hence provide basis for Industry 4.0 solution development. One such simulation approach, DES has been applied in several cases to optimize, decide and improve the operational performance of numerous production lines. However, limited models are available in literature to support battery production lines and therefore, in this paper, discrete event simulation is utilized to model a battery module assembly, with the intention to *i*) understand the best practice for scale-up of pilot line to full scale production, *ii*) comprehend the challenges imposed by the system configuration during scale-up, *iii*) integrate the principles of scale-up with scheduling policies and *iv*) compare potential scale-up strategies.

3. Methodology

The research focus is on the pilot line battery module assembly and their subsequent scale-up policies. Pilot production lines serve as a transition phase from concept development to full-scale production, wherein the validation of product and process is carried by pilot runs [8]. The plethora of data available from these production lines can serve as input for efficient identification of potential disturbances, comparison of scale-up strategies, fine tuning of process parameters and predictive maintenance of bespoke machines. Figure 1 shows the proposed methodology which is explained in detail further.

3.1. Overview

From Figure 1, the operational performance of an initial virtual representation of the system is analysed in a DES software. Data obtained from the pilot production line, such as the process and setup time, available space, material feeding etc., is used in this stage to develop the model. This initial model is subjected to two scale-up strategies. Strategy 1, which will henceforth be referred to as INC referring to the increase in functionality, involves decreasing the process time by increasing the efficiency and performance of various machines such that the required demand can be met. Strategy 2, which will henceforth be referred to as REP referring to replication, involves addition of stations with similar functionality in parallel, thereby increasing resources to meet the demand. Two dispatching rules are assigned to INC and REP. One of the rules is First In First Out (FIFO) and the other is Shortest Processing Time (SPT). This combination generates four scenarios as follows

- S1 INC with SPT
- S2 INC with FIFO
- S3 REP with FIFO
- S4 REP with SPT

The criteria for comparison of the scenarios depends upon the throughput. The threshold is set as x products of type A and y products of type B; scenarios or replications of scenarios that result in throughput less than this number are not considered for comparison. From literature, two statistical techniques that are used to compare the scenarios generated in DES are Ranking and Selection and Multiple Comparison techniques [18]. In this study, Multiple Comparison Procedures are identified as the most suitable approach as they provide information about the differences between the different scenarios in comparison. Therefore, the selected scenarios are subjected to statistical analysis as seen from Figure 1; MANOVA and ANOVA are explained in sections 5.1 and 5.3 respectively following which the results are discussed.

It is to be noted that, although the same methodology might be applicable to different production systems, the results and behaviour obtained and discussed in this study are a consequence of the initial system configuration in consideration. The following sections explain the reasoning behind selection of the policies and rules for scale-up and scheduling.

3.2. Scale-up

The concept of pilot production line has been briefly discussed earlier. Pilot production phase is usually followed by a ramp-up and/or full-scale production. Therefore, it is essential that the most suitable strategy for full-scale production to be identified well in advance to reduce the time to market. Virtual engineering toolsets, in particular, discrete-event simulation models are capable of providing support in this decision making. Based on the two principles provided by [9], station replicating and increasing functionality have been chosen. In REP, the stations that are over-utilized are identified by running experiments in the software and additional stations that serve the same functionality are added to the system. This, however, results in an increase in the number of operators if the added stations are manual. Additional floor space is required for this expansion as well. Therefore, more operators are assigned to the stations inside the cyber model. Although there is cost associated with this, it has not been quantified in this research. The comparisons have been made from an operational behaviour point of view. On the other hand, in INC, the identified stations are assessed for potential functional improvements and by increasing the functional characteristics, the new demand is met without any addition of stations. The process and setup times of the stations in the software model are reduced to represent this increased functionality. However, this is not discussed in detail in this study since this required immense amount of data regarding the details of the proposed improvement which is hard to establish in the concept stage. It is to be noted that the number of stations remains the same as the initial model, hence there is no necessity to add more operators to the system. The next section explains the scheduling policies adopted.

3.3. Scheduling and dispatching rules

In a production environment, scheduling and sequencing of jobs can be done at various phases. Static schedules are generated at the start of the production run and are not changed, whereas, dynamic schedules are generated whenever a disturbance occurs during production that demands a change in the existing schedule. Several dispatching rules are applied during production scheduling to select products according to certain established priorities. For the purpose of this study, two dispatching rules, namely First In First Out (FIFO) and Shortest Processing Time (SPT) are selected and their combination with the above-mentioned scale-up principles generates the four scenarios which will be discussed in the following sections.

4. Case study initial model

Battery module assembly is performed prior to the pack assembly, wherein cylindrical batteries are arranged in a pre-determined pattern to obtain the required energy and power. During this process, various components for module framework, cooling system, electrical connections etc. are fitted. A schematic diagram of the initial system configuration is shown in Figure 2. The key features of the system in discussion are as follows. The cylindrical 18650 Li-ion cells that are assembled, have to be accessed from both directions, the top and the bottom, to achieve the joining process. Therefore, there is need for a reorientation operation after which the joining process has to be repeated. In order to realise this, the conveyor system is provided with a loop as shown in Figure 2. The two sources A and B generate the two variants respectively.

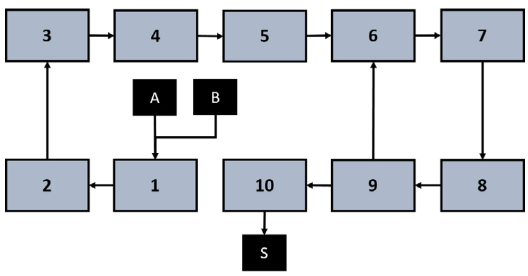


Fig. 2. Initial model representation.

Table 1 shows the process sequence for the two product variants that are assembled on the line. It can be seen that, when a product variant does not need processing at a station, it can bypass the station with the help of RFID (Radio Frequency IDentification) tags. For instance, product *B* does not need to be plastic welded and hence it bypasses station 5. Similarly, product *A* does not require ultrasonic welding and hence it does not need to be operated at station 6. The aspect of the case study which needs to be highlighted is the presence of the loop/shuttle from station 9 to station 6 which provides some routing flexibility. Product *B* does not have the need to travel the loop, however, product *A* is subjected to pulse arc welding on both cell terminals and hence travels through the loop and gets processed in station 8 twice. The production system has a throughput of 55 products per day with automated stations of 2, 6 and 8. Six operators work on 7 manual stations and travel to a station on a requirement basis.

Table 1. Process sequence for the two product variants.

Station number	Product A	Product B
1	Assemble carrier tray	Assemble carrier tray
2	Cell loading and testing	Cell loading and testing
3	Install cooling system	Inspection
4	Assemble top tray	Install busbar
5	Plastic weld housing	-
6	-	Ultrasonic wire bonding
7	Install busbar	Install busbar
8	Pulse arc welding	Pulse arc welding
9	Assemble insulation cover (top)	Weld inspection
10	Assemble insulation cover (bottom)	Assemble cover plate

4.1. DES model parameters

The scale-up model creation process exacts various parameters to be defined. The new demand is assumed to be twice that of the initial one and this is reflected by an increase in the inter-arrival time for products *A* and *B*. The product mix ratio is 70% product *A* and 30% product *B* and batching is not considered. Each station has a setup time which will be considered when product type changes. A warm-up time of 10000 seconds is considered to allow the system to reach steady state for performing statistical analysis. The simulation is run based on a shift time of 28800 seconds and stochasticity is introduced into the model using statistical distributions. For instance, mean time to failure values are modelled using the exponential distribution. 100 replications are performed for each of the scenarios. The presence of the loop/shuttle in the model can result in unprecedented behavior of the system with respect to product flow time. However, no buffer stations are considered in the model. A schematic representation of the REP scenarios (S3 and S4) is shown in Figure 3. Since the INC scenarios (S1 and S2) do not have a change in their configuration they look identical to the initial model shown in Figure 2. Although, there are several performance measures that arise from quality and operational domain, the key performance indicator that is considered for this study is the mean flow time of products A and B.

5. Results and discussion

A comparison of the operational performance of the four scenarios is performed by Multivariate ANalysis Of VAriance (MANOVA) and ANalysis Of VAriance (ANOVA) to statistically identify the existence of significant difference between the scenarios. For both tests, the four scenarios represent the independent variable.

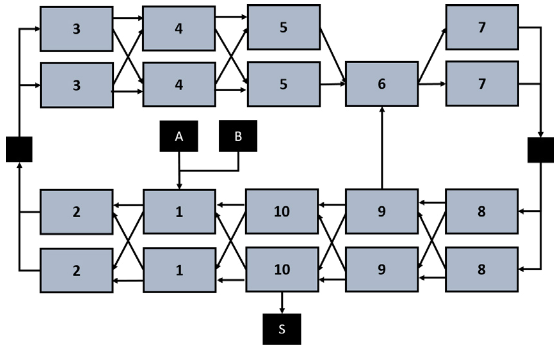


Fig. 3. Schematic representation of scenarios 3 and 4.

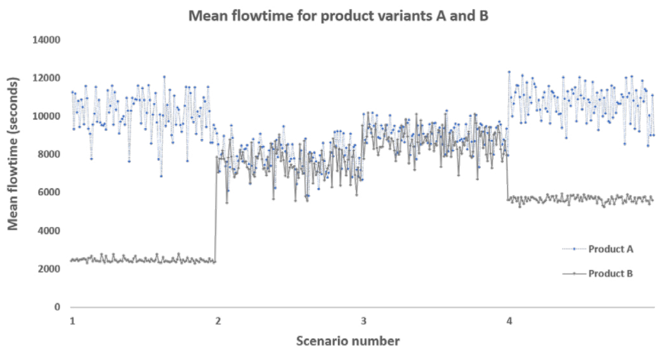


Fig. 4. Mean flowtime for the two product variants.

5.1. MANOVA testing

The two dependent variables required for MANOVA are the mean flow time for products *A* and *B* respectively. There are several assumptions that need to be satisfied to run the tests and this was performed in SPSS. Few assumptions were violated, however, it is expected that the effect of this violation will be negligible due to the sample size considered. Subsequently, Pillai's trace values in the multivariate test results were considered for analysis. P-value less than the significance level of 0.001 is obtained.

5.2. MANOVA results

The null hypothesis H_0 in MANOVA states that all the scenario means are equal

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 \qquad (1)$$

Where μ_1, μ_2, μ_3 and μ_4 are the means of the respective scenarios. Since the p value is less than the significance level, null hypothesis is rejected and at least one set of means is significantly different from another. To understand more about this difference, a multiple comparison procedure called Tukey's Honest Significant Difference (HSD) test is considered. A comparison of mean flowtime for products *A* and *B* for four scenarios is shown in Figure 4. Figures 5 and 6 show the results obtained from Tukey's HSD test. From Tables 2 and 3, the values in the subset column represent the mean flowtime for the scenarios and it can be seen from both tables, that none of the scenarios share a subset; the mean flowtime of all the scenarios are significantly different from each other for both products.

Table 2. Homogenous subset output for MANOVA testing of product A

Scenario No.	Subset 1	Subset 2	Subset 3	Subset 4
2	7842.24			
3		9053.77		
1			10096.25	
4				10708.55

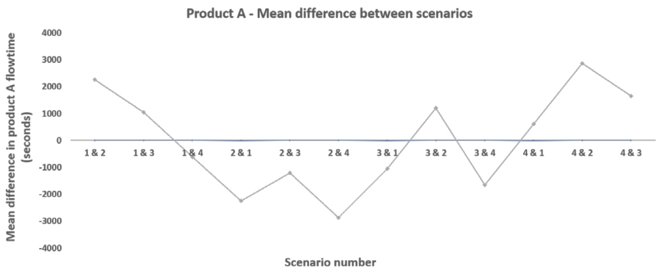


Fig. 5. Mean difference between scenarios for product A flowtime.

Table 3. Homogenous subset output for MANOVA testing of product B

Scenario No.	Subset 1	Subset 2	Subset 3	Subset 4
1	2453.68			
4		5639.33		
2			7473.13	
3				8646.28

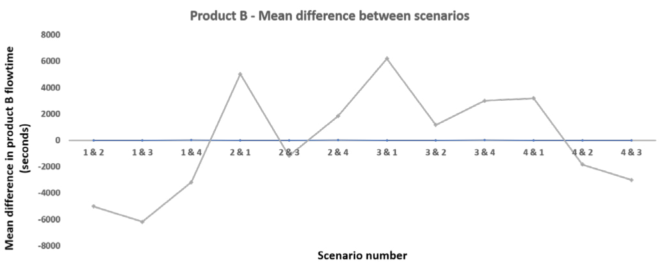


Fig. 6. Mean difference between scenarios for product B flowtime.

5.3. ANOVA testing

The MANOVA test was conducted considering the two product flow times as different dependent variables. Although, the results provide valuable data, the effect of combining flowtime of both products is not perceivable from the obtained results. Hence ANOVA was performed by considering the flowtime as one dependent variable by adding the mean flowtime of products *A* and *B* for each replication of each scenario. Assumption tests were conducted identical to the previous case. P-value of less than 0.001 was obtained and hence the null hypothesis that the scenario means are equal can be rejected.

5.4. ANOVA results

The rejection of null hypothesis implies that at least one set of means is significantly different from another. The total flowtime (mean flowtime *A* + mean flowtime *B*) for the 100 replications in each scenario is shown in Figure 7 and the mean difference between the scenarios is shown in Figure 8. From Table 4, the total mean flowtime for the scenarios are in different subsets; the mean flowtime for all four scenarios are significantly different from each other.

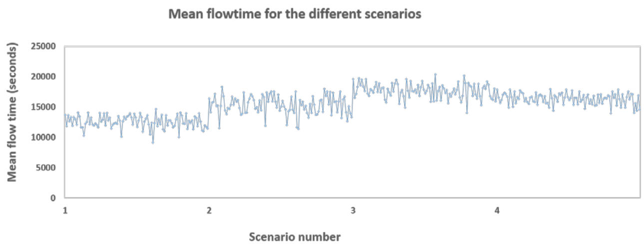


Fig. 7. Total mean flowtime for different scenarios.

Table 4. Homogenous subset output for ANOVA test

Scenario No.	Subset 1	Subset 2	Subset 3	Subset 4
1	12549.94			
2		15315.38		
4			16347.88	
3				17700.06

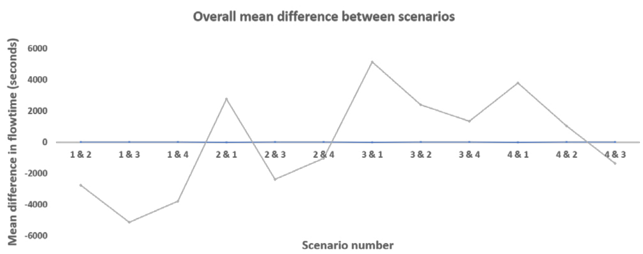


Fig. 8. Mean difference between scenarios for total mean flowtime.

5.5. Discussion

Comparisons of the mean flowtime of products *A* and *B* as seen from Figure 4, reveals that the flowtime of product *B* is influenced heavily by the type of dispatching rule considered. From Tukey's test (Figure 5), the mean difference between S2 and S4 is approximately 3000 seconds. Therefore, INC with FIFO dispatching rule allows product *A* to be assembled much faster than other scenarios. It is to be noted that REP with SPT dispatching rule increases the overall processing time of product A. On the other hand, INC with SPT dispatching rule reduces the mean flow time of product *B* considerably, whereas REP with FIFO increases the mean flowtime of product B. The comparisons performed so far, have considered the flowtime of the product variants separately. However, considering the total flow time of products *A* and *B*, from Figure 7, it is evident that INC with SPT dispatching rule reduces the total assembly time. Another trend that can be identified is the relative increase in flowtime for scenarios adopting the REP scale-up strategy when compared to INC scale-up strategy. Therefore, it is safe to assume that for the considered performance measure, initial system configuration, product variants and processing times, a scale-up strategy which involves improving performance of machines/stations by increasing their functionality integrated with the SPT dispatching rule provides good results. The proposed approach can be useful for decision making with the caveat being the inability to compare the prediction results from a cyber model with actual results from a physical model.

6. Conclusion and future work

In this study, two distinct scenarios for scale-up have been proposed. However, a hybrid strategy that combines the INC and REP could possibly be considered for future purposes. The data regarding processing time has been obtained from the pilot line for creating the DES models. However, quality data that can be inferred from the setup time change has not been considered for analysis. Moreover, there is possibility to feed data to machine learning algorithms to better predict scale-up strategies. In this research, only two of the many available dispatching rules have been compared. There is also potential of considering scheduling at different phases of production. For instance, when a disturbance such as machine breakdown occurs, a change in dispatching rule to reduce the effect of disturbance could be considered. Throughout the study, a particular initial system configuration has been considered, however, many such experiments can be conducted using different initial system configurations of battery module assembly and the obtained data could help predict best practice scale-up strategy for full-scale production.

This research highlights the importance of battery manufacturing and assembly in current industrial scenarios. Consequently, best practice for development and assembly of battery modules and packs is the need of the hour. Therefore, it is essential to validate products and processes in pilot production lines, which ultimately must be scaled-up for full scale production. The profuse quantity of data generated in such lines can support the creation of virtual models to understand scale-up strategies. Data regarding the operational performance and routing of the stations is fed into DES model and integrated with scale-up policies and dispatching rules to generate four different scenarios. The performance of the scenarios is compared statistically to support decision making. Although the proposed methodology is implemented in a system that assembles battery modules, it is possible to extend this approach to other manufacturing systems. It is, however, necessary to check the availability of sufficient space for adding new processing units, the possibility of increasing the functionality of a machine, etc. prior to the implementation. Additionally, the potential benefits of this implementation to a specific application or scenario, could be ascertained with the help of experiments.The authors believe that this research study proposes a methodology to i) guide good practice scale-up from pilot production line, and ii) develop cyber-physical architecture at the pilot line level, by using DES as a tool for decision making and guiding the smooth transition from pilot line to full-scale production.

Acknowledgements

The authors acknowledge the support of Innovate UK through the AMPLiFII project (ref: 102490), and UK EPSRC through KDCM (ref: EP/K018191/1).

References

[1] Michela Longo, Wahiba Yaïci, and Dario Zaninelli. Team play between renewable energy sources and vehicle fleet to decrease air pollution. *Sustainability*, 8(1):27, 2015.
[2] Ralf Kalmbach, Wolfgang Bernhart, P Grosse Kleimann, and Marcus Hoffmann. Automotive landscape 2025. *Opportunities and challenges ahead (= Studie von Roland Berger Strategy Consultants)*, 2011.
[3] Dirk Uwe Sauer. Dont wait for the next battery-cell generation. *ATZelektronik worldwide*, 11(5):72–72, 2016.
[4] Adrian Kölmel, Anna Sauer, and Gisela Lanza. Quality-oriented production planning of battery assembly systems for electric mobility. *Procedia CIRP*, 23:149–154, 2014.
[5] Ashkan Negahban and Jeffrey S Smith. Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, 33(2):241–261, 2014.
[6] Mohsen Jahangirian, Tillal Eldabi, Aisha Naseer, Lampros K Stergioulas, and Terry Young. Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203(1):1–13, 2010.
[7] Malte Schönemann. *Multiscale Simulation Approach for Battery Production Systems*. Springer, 2017.
[8] Henrik Almgren. Pilot production and manufacturing start-up: the case of volvo s80. *International Journal of Production Research*, 38(17):4577–4588, 2000.
[9] G Putnik, A Sluga, H ElMaraghy, R Teti, Y Koren, T Tolio, and B Hon. Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap. *CIRP Annals-Manufacturing Technology*, 62(2):751–774, 2013.
[10] Michael Levin. *Pharmaceutical process scale-up*. CRC Press, 2001.
[11] Ahmed M Deif and Waguih ElMaraghy. Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system. *The International Journal of Advanced Manufacturing Technology*, 32(5):557–562, 2007.
[12] Ahmed M Deif and Hoda A ElMaraghy. Assessing capacity scalability policies in rms using system dynamics. *International journal of flexible manufacturing systems*, 19(3):128–150, 2007.
[13] Wencai Wang and Yoram Koren. Scalability planning for reconfigurable manufacturing systems. *Journal of Manufacturing Systems*, 31(2):83–91, 2012.
[14] Lothar Seybold, Marcin Witczak, Paweł Majdzik, and Ralf Stetter. Predictive fault-tolerant control for a battery assembly system. *IFAC-PapersOnLine*, 48(21):476–483, 2015.
[15] Alexander Tornow, Robert Graubohm, Franz Dietrich, and Klaus Dröder. Design automation for battery system variants of electric vehicles with integrated product and process evaluation. *Procedia CIRP*, 50:424–429, 2016.
[16] Sha Li, Hui Wang, S Jack Hu, Yhu-Tin Lin, and Jeffrey A Abell. Automatic generation of assembly system configuration with equipment selection for automotive battery manufacturing. *Journal of Manufacturing Systems*, 30(4):188–195, 2011.
[17] Achim Kampker, Kai Kreisköther, Martin Hehl, Sebastian Gillen, and Maximilian Rothe. Discrete event simulation approach considering scalable systems and non-expert users in the early phase of production planning for electric powertrains. In *Industrial Technology and Management (ICITM), International Conference on*, pages 139–144. IEEE, 2017.
[18] James R Swisher, Sheldon H Jacobson, and Enver Yücesan. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):134–154, 2003.

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 81 (2019) 962–967

www.elsevier.com/locate/procedia

52nd CIRP Conference on Manufacturing Systems

A Framework for Pilot Line Scale-up using Digital Manufacturing

Malarvizhi Kaniappan Chinnathai*, Zeinab Al-Mowafy, Bugra Alkan, Daniel Vera, Robert Harrison

Automation Systems Group, WMG, University of Warwick, CV4 7AL, West Midlands, UK

* Corresponding author. Tel.: +44 7831001684. *E-mail address:* m.kaniappan-chinnathai.1@warwick.ac.uk

Abstract

Pilot lines are essential test-beds for process and product validation before the establishment of production lines. However, there is a lack of well-defined methodology for pilot line scale-up. To better support this transition, Virtual Models can be integrated with Discrete-Event Simulation (DES) models for potential production-line configurations. However, the validation of the developed models is hardly possible due to the absence of a physical counterpart. Therefore, this paper proposes a framework to increase the accuracy of the DES scale-up models with Virtual Modelling tools and Ontology. Subsequently, a test-case is used to explain the concept.

Keywords: Digital manufacturing; Pilot line; Discrete-Event Simulation;

1. Introduction

The challenges faced by today's manufacturing industries are fueled by the increased product variety, rapid changes in technology, reduced time-to-market and shortened product life-cycle [1]. To cope up with the reduced time-to-market, firstly, it is important to achieve reduced time-to-volume i.e. to advance from the conceptual phase to full-volume production with increased thrust. During the conceptual phase, it is not uncommon for manufacturing industries to produce prototypes for purposes such as testing and validation of product, process and resource design. As it is crucial to achieve a successful transition from design phase to time-to-volume, it is essential to use pilot lines to identify potential disturbances prior to commissioning of the line [2]. A myriad of issues actually arise in early design phases and are not detected until commissioning; anticipating these issues before commissioning of production lines can ensure successful upscaling that can provide a competitive market advantage [3,4,5]. A successful scale-up project significantly reduces the time-to-market which consequently enables the industry to secure more revenue by dominating the market [5]. Although a plethora of articles have been published pertaining to the identification and management of disturbances and issues that could be faced during the up-scaling procedure [2,6,7], there is still lack of a robust methodology to enable the scale-up process in a smooth way. To support the transition from planning phase to full-volume, however, simulation and modelling is identified as one of the enabling technologies [8,4,6].

The concept of digital manufacturing has previously been found to support the manufacturing system and detect potential disturbances and issues affecting the line [3]. For this purpose, there are several commercial tools available, however, the underlying principles and techniques on which they function varies widely. In this paper, two simulation methods i.e. Virtual Modelling and Discrete-Event Simulation (DES) are identified and integrated with pilot line data to support the scale-up process. For several years, DES has been widely used for

10.1016/j.procir.2019.03.235

supporting manufacturing industries [9]. DES finds use in identifying and analysing potential scale-up scenarios with input data from the pilot line [8]. However, as a standalone tool, DES does not have the capability to analyse the feasibility of the modelled scenarios of the future manufacturing line; this could potentially result in a situation where the solution offered through simulation might not actually be possible to realise. In specific, the assumption of station processing time values of potential production line models due to the absence of real system could lead to misleading results. To overcome this drawback, DES software module can be integrated with a Virtual Modelling software that models the kinematics, geometry and the logical behavior of the workstation resources. Commercially available PLM suites offer this capability to integrate multi-level software modules, but their implementation, training and license cost is exacting [10]. Moreover, there is requirement for the integration of heterogenous software tools within the overarching concept of digital factory [11].

1.1. Summary

From the above-mentioned discussion, the key points can be summarized as follows: i) the use of digital software modules can support the upscaling phase ii) DES software, if used as a standalone module, is not smart enough to identify whether the assumed station process times for future scenarios is feasible or not and iii) the integration of heterogenous digital software modules is aligned with the concept of digital factory.

1.2. Key contribution

Therefore, the core benefits of this paper are twofold i) proposal of an approach for integration of data from Virtual Modelling tools with an ontology software to calculate station process time such that the accuracy of the DES models are improved and ii) supporting the transition from pilot line to full-scale, subsequently shortening the time-to-market.

2. Literature review

2.1. Digital manufacturing

The notion of using simulation tools for manufacturing is not a new one. The software tools differ in their method and level of detail with which they model the system. This review briefly touches on production line modelling, namely Discrete-Event Simulation and workstation modelling referred to as Virtual Modelling.

2.1.1 Difference between DES and Virtual Model

Amongst the available tools for modelling the production line for operational research, Discrete-event Simulation is identified to be the most popular one [12,13] . Conventionally, DES is used for operational phase analysis, but its benefits can be exploited during the early stages of production as well [14]. The benefits of employing DES during early design stage include layout planning, material handling design, etc. and during the operational phase for scheduling and operational policies, and real-time control. However, in DES, analyses are performed by modelling the system with higher level of abstraction with the process and workstation level detail not included in the model; the focus is on detailing the production line and product flow. On the other hand, Virtual Modelling tools are used to model and analyse the system at the workstation or machine-level. They encompass information about the kinematic model (geometry and joint), behavior model (transition and states) and the reference coordinate system [3]. Moreover, they can be used to analyse ergonomics, collision detection, validation of PLC codes and design planning [11].

2.1.2 Benefits of integrating DES and Virtual Model

The primary benefit of integrating the Virtual Modelling tool with DES is to support the production-line level model in DES with the workstation-level details such as station processing times, breakdown information, robot motion time, human performance modelling, energy consumption and layout modifications [15,11].

Several commercial PLM suites have software modules that perform Virtual Modelling and DES. Additionally, these modules are present on an integrated platform that supposedly allows the sharing of data in a seamless way and thereby realizing the integration of Virtual Models and DES models. Although PLM tools have this capability, the tools are not affordable for SMEs due to i) cost of training and license ii) cost of changing infrastructure to adapt to the PLM environment iii) replacing any existing specialized software with the PLM toolset and the cost of implementation of PLM [16,17]. Moreover, from the view of digital factory, it is difficult to integrate PLM tools with heterogenous software and databases [18].

2.2. Summary

An analysis of articles about digital manufacturing indicates the following: i) quantification of the benefit of integrating heterogenous digital tools and ii) the lack of knowledge on the benefits and the procedure for integration of Virtual Modelling with DES to successfully support smooth transition from planning to full-volume. Therefore, in this research study, the authors propose an approach to support the transition from pilot phase to full-scale production by leveraging the integration of Virtual Modelling tool and DES.

3. Methodology

The research presented in this paper is primarily aimed at upscaling of assembly systems. The core idea of this research article is to share relevant workstation data from Virtual Modelling tool and the existing pilot line with an ontology tool to generate a list of station process times. The station process time data is necessary for ensuring that results of DES are

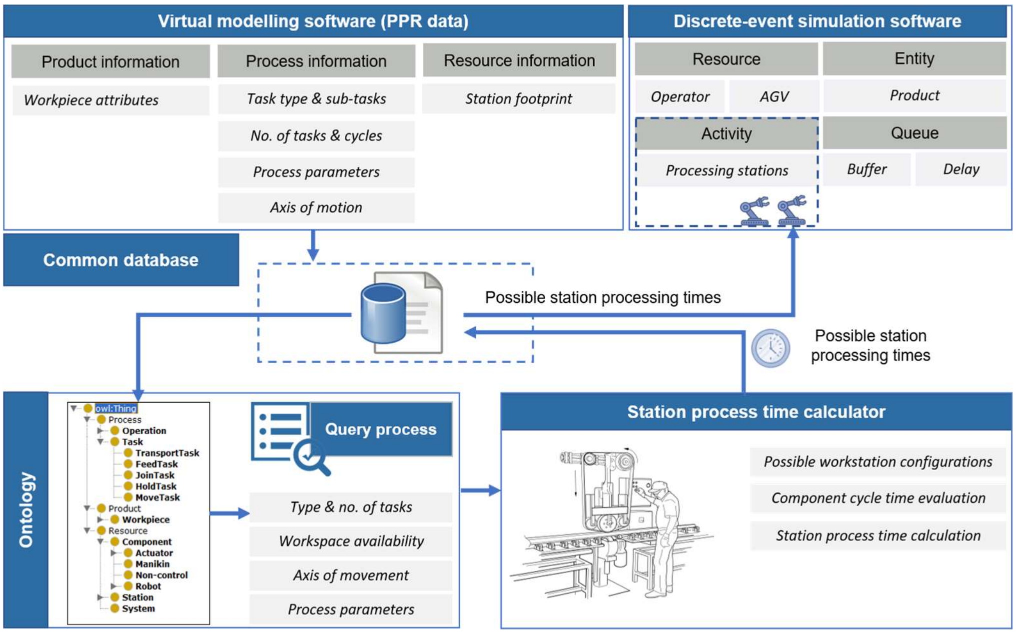


Fig. 1. Methodology.

realistic. The assumption of the process times could lead to situations where the models have workstation process times that might be too high or too low for the considered workstation configuration which could adversely affect the simulation results. The information sharing between the different tools is achieved using common database. From Fig 1, the common database model is a centralized database model which has been created in a way to support the integration of different software modules. In this paper, a common database scheme was designed to store the Virtual Model information in 3 tables: Product, Process and Resource. Each table has columns to represent the considered parameters and their respective IDs. A relational mapping between the Virtual Model and ontology classes facilitates provision of data for query and inserts the calculated results back into the tables. The common database model allows automating the integration between the virtual modelling software, ontology software and DES. The key concepts of the methodology can be explained as i) integration of Virtual Model with Ontology and ii) station process time calculator.

3.1 Integration of Virtual Model with Ontology

Virtual Modelling tools have capability to store information about product, process and resource at the workstation level; this information can also be shared with other tools. Within the context of this paper, a manufacturing resource comprises of system, station and component with increasing level of detail. A component is defined as the basic unit of a system that can be sub-divided into elements [19]. As an example, a robot can be considered as a component and the drives and motors of this component are the elements. The data from the existing pilot line serves as the crucial input for creating the Virtual Model. Table 1 shows the data intended to be used by the ontology model.

It is important to note that a significant proportion of this data is obtained from the existing pilot line. The task types are

Table 1: Input data for ontology

Data type	Description
Workpiece attributes	Product features that are necessary to filter system resources that can perform the assembly.
Task type	Five types of tasks are considered: move, hold, transport, feed and join.
No. of tasks	The number of tasks that are performed in a workstation.
No. of cycles	The total number of cycles to perform an operation at the workstation.
Sub-tasks	The sub-task corresponds to the specific actions that are executed to achieve a task
Process parameters	Process parameters represent the accuracy, repeatability, force requirement, torque etc, for carrying out an operation.
Station footprint	The dimensions of the workstation that helps determine the available space to configure the workstation.
Axis of motion	The degree of freedom of the 'current resources' that are used in the pilot line/virtual model

decomposed as shown in Table 1, however, the inspection and testing operations are not included within the scope of this research [20]. The axis of motion of the system resources in the existing line essentially enables removing components that have less axis of motion from future workstation configurations. This helps eliminating options that have less productivity than the existing components in pilot line, with the underlying assumption that an increase in the axis of motion, i.e. from a 3D gantry to six-axis robot, increases the productivity.

Ontology is defined an explicit specification of a conceptualization and the development of ontology enables the sharing of common understanding of a domain between people and application systems [21]. The idea behind the use of the ontology model is that the process parameters, task type,

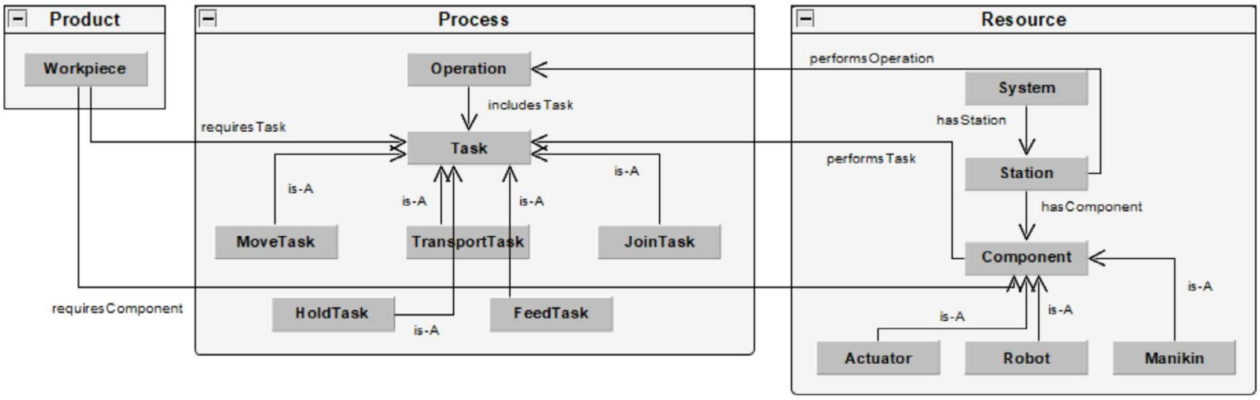


Fig. 2. Ontology definition in protégé.

number of tasks, axis of motion and station footprint data can be used to filter an available catalogue of assembly equipment to find those that meet the requirements. In this paper, the software protégé [22] is used to define the ontologies and three classes, namely product, process and resource as shown in Fig. 2. Three types of assembly components are considered within the scope of this paper: actuator, manikin and robot. Although, the proposed approach is suitable for all the assembly components and task type considered, this research article will focus on the 'move components' and 'hold components'. A catalogue of components can be created, either in the database or protégé, which consists of potential assembly equipment that are at the disposal of the industry.

Following the definition of ontology, a query operation using SPARQL on the generated equipment list will enable identifying the components that can i) perform the required number and type of tasks ii) fit within the available workspace iii) able to satisfy the process parameters and iv) have the required axis of motion. From the resulting list of components, the next step is to calculate processing time of the workstation when the selected components are used. Essentially, the station process time is expected to vary with component and the method of calculation is explained in the next section.

3.2 Station process time calculator

The station process time calculator (Fig 3) considers the type of task, either 'move' or 'hold', and the selected components for each are listed as $[M_1, M_2 \ldots M_n]$ and $[H_1, H_2 \ldots H_k]$, where 'n' is the total number of selected components for 'move' task and 'k' is the total number of selected components for 'hold' task. From Fig 3, the 'sub-task level' shows the sub-tasks performed for a pick and place operation, wherein two tasks 'move' and 'hold' are involved. The information in the sub-task level are acquired from the virtual model. The motion times for the sub-tasks of each of components $[M_1, M_2 \ldots M_n]$ and $[H_1, H_2 \ldots H_k]$ are calculated with data from different sources: physics-based model of the component that can calculate the motion time, experience-based motion time, machine-learning from previous projects, motion time from component datasheet or from virtual modelling software. Additionally, it is important to understand the distance that actuators are displaced by during the 'move' and 'hold' tasks

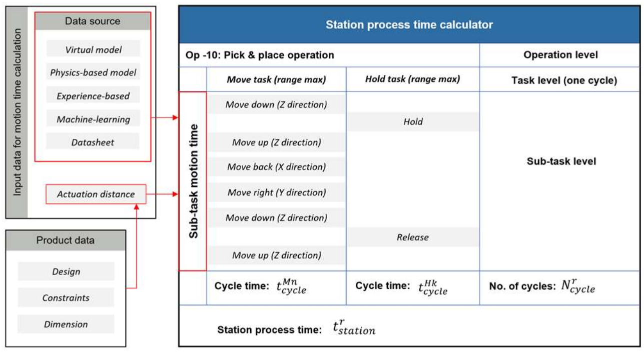


Fig. 3. Station process time calculation.

to calculate the motion time. Essentially, the product dimensions, design and constraint details can be translated to dimensional values in the Virtual Model that provides the necessary data for actuator displacement distance. The letters

$$t^{M_n}_{cycle} = \sum_{z=1}^{j} t^{z}_{motion,\,M_n} \quad (1)$$

$$t^{H_k}_{cycle} = \sum_{t=1}^{m} t^{t}_{motion,\,H_k} \quad (2)$$

'j' and 'm' represent the total number of sub-tasks for the 'move' task and 'hold' task respectively. The motion times of the sub-tasks for component M_1 is represented as $[t^1_{motion,M1}, t^2_{motion,M1} \ldots t^j_{motion,M1}]$ and the motion times of sub-tasks for component H_1 is represented as $[t^1_{motion,H1}, t^2_{motion,H1} \ldots t^m_{motion,H1}]$. Similarly, the motion time of the sub-tasks for each of 'n' components for 'move' task and 'k' components for 'hold' task can be calculated. Following the calculation of motion time, the cycle time for the components performing 'move' task and 'hold' task can be calculated using Equations 1 and 2 respectively. To find the total cycle time, t_{cycle}, the cycle time for the 'move' task and 'hold' task should be added together. Therefore, each of the component performing 'move' task will be added with each of the component performing 'hold' task that will result in n*k cycle time values. This is then multiplied with the total number of cycles per operation, N^r_{cycle}, to obtain the station processing time, $t^r_{station}$. It is assumed that each station performs one operation and the total number of stations is represented as $N_{station}$ and 'r' is an index that represents the station number. This list of station processing times for each operation performed in the production line is stored in the common database and readily available for performing analyses in DES. Typically, in DES software, the station process time is a parameter that does not have any rules to determine whether the time is a feasible one or not. Integration with the database allows only the verified time values to be used in DES and subsequently improves the accuracy of the model. There is a choice of different station process time values stored in the database for each workstation and it provides the user the flexibility to choose process time according to certain criteria.

4. Case study

The proposed methodology is applied to a battery module assembly case. The station that is considered is the 'cell loading station', where '18650 battery cells' are picked up by a three-axis gantry with vacuum gripper and placed in a battery module. The station model is created in a virtual modelling toolset called 'VueOne' developed in the Automation Systems Group, University of Warwick. The software has two platforms that enable creation and definition of the component and station. The components such as gripper and gantry unit are the actuators that are associated with logical behavior. On the other hand, the station frame is considered as non-control component due to the absence of a logical behavior.

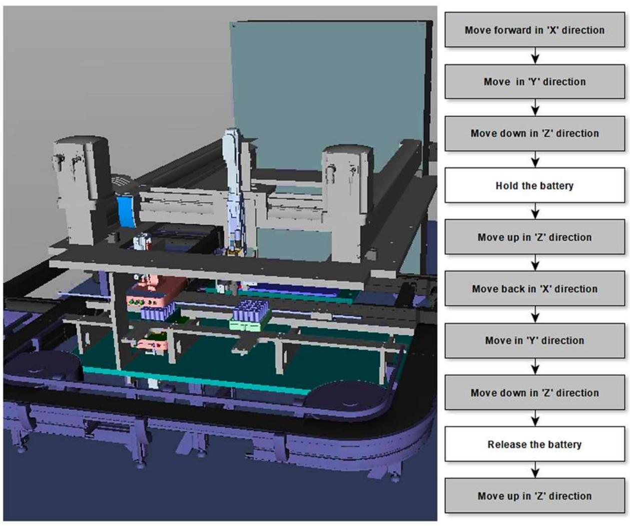


Fig. 4. Model in VueOne with process sequence.

The model that is created in VueOne and the process sequence of the sub-tasks are shown in Fig. 4. The coloured boxes represent the 'move' sub-tasks and the white boxes represent the 'hold' sub-tasks. In this example, the number of 'move' sub-tasks 'j' equals 8 and the number of 'hold' sub-tasks 'm' equals 2. The data from Virtual Model are represented in Table 2.

To demonstrate the methodology, potential components were queried from the VueOne component library to identify those components that meet the requirements in Table 2. For the 'move' task, a total of nine gantries were queried and four were found suitable. For the 'hold' task, a total of 53 grippers were queried and nine were found suitable.

4.1 Cycle time calculation

The motion time for the 'move' task is calculated for the four selected gantries. The gantries should perform 'eight' sub-tasks, the motion time of which is obtained from the gantry datasheets. A summation of the motion time results in four cycle time values. Similarly, the 'hold' task comprises of 'two'

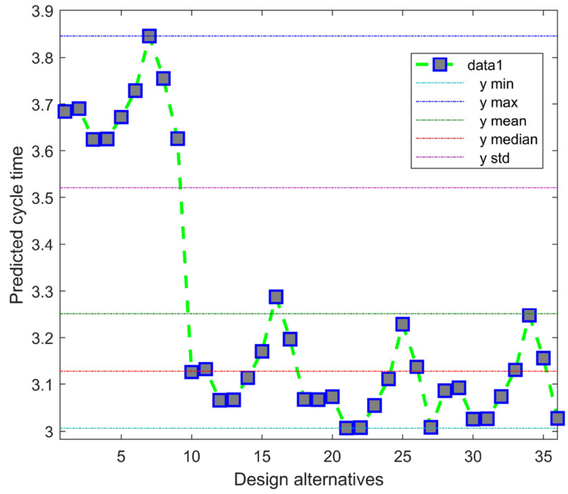


Fig. 5. Case study results.

sub-tasks, the motion times are calculated from the gripper datasheet and summed up to obtain nine cycle time values. This results in 'four' cycle time values for the gantries and 'nine' cycle time values for the grippers. The total cycle time is calculated by adding the 'move' and 'hold' cycle time values for identified components which results in a total of '36' cycle time values which are illustrated in the plot in Fig. 5. This provides decision support for choosing the best combination of components depending on the cycle time requirements. For example, from Fig. 5, the combinations 21, 22 and 27 have very less cycle time values and could be considered as candidates for the new workstation configuration. Since for considered case, the operation has 100 cycles, the cycle time values are multiplied by 100 to obtain the station processing time values.

Table 2: Data from Virtual Model

Data type	Values		
Working range required (in mm)	X	Y	Z
	750	450	300
Workspace availability required (in mm)	2000	1500	1000
Positioning accuracy required (in mm)	0.5	0.5	1
Number of cycles	100		
Axis of motion	3		
Drive type	Electric		
Payload (in gram)	45		

4.2 Integration with DES

The resulting station process time values are stored in the common database. The line level model of the pilot line is created in DES using the commercially available tool provided by Lanner group called 'Witness'. The pilot line consists of eight workstations and the process time for seven workstations are assumed, whereas for workstation 1 which is the test case of cell loading station, the process time values are retrieved from the common data base using 'in-built' functions available in 'Witness'. Thereby, the cell loading station has more realistic process-time values that are obtained by the integration between VueOne and protégé. The station process time data can be linked with other decision supporting criteria such as cost, machine breakdown information etc. for multi-criteria decision making.

5. Future work and discussion

The proposed methodology is demonstrated for a pick and place operation, but it can be extended to other types of operations as well. Although the primary focus in this research was calculation of the cycle time of 'actuators' like gantry and grippers, the methodology is applicable for robots and digital human models. Additional work will be done to apply the proposed methodology to robotic stations and manual workstations. The methodology primarily targets improving the functionality of the existing stations by replacing the components. However, the changes in layout configuration of the workstations are not considered. The authors plan to perform further analysis in DES by incrementing the station quantity and performing layout modifications and integrating it with the workstation level analysis achieved in this paper. This will provide a holistic view of the scale-up from workstation as well as production line level. One major limitation of the approach is that the motion time values from data sources in Fig. 3, may not be accurate. Moreover, for simple processes the calculations for cycle and process time performed in this paper can be approximated to be close to the real, however, for complex processes this may not be the case. More work needs to be done in this area to enrich the data sources in Fig. 3 with better and realistic component motion time values by employing machine learning techniques.

6. Conclusion

This article presents an approach to demonstrate the integration of a virtual modelling tool with an ontology model to calculate the station process time. Additionally, the common database stores the station process time which can be accessed by the DES software as and when necessary. This essentially improves the accuracy of the DES model with more realistic time values that are significant to perform meaningful production line analysis. Therefore, the integration of workstation level model using Virtual Modelling software with a line-level model using DES software is proposed to support the upscaling process. It is envisioned that the decision-support provided by the methodology can significantly reduce the time-to-volume and ultimately result in cost and time savings.

Acknowledgements

The proposed approach is part of ongoing research studies through the project AMPLiFII 2 (Innovate UK). The primary objective of the research is to guide the scale-up of the battery module assembly facility, present at WMG, University of Warwick, by leveraging the pilot line data and integration of digital manufacturing tools.

References

[1] ElMaraghy HA. Flexible and reconfigurable manufacturing systems paradigms. Int. J. Flex. Manuf. Syst., 17(4): 261–276, 2005.
[2] Almgren H. Pilot production and manufacturing start-up: The case of Volvo S80. Int. J. Prod. Res., 38(17): 4577–4588, 2000.
[3] Liu Z, Suchold N and Diedrich C. Virtual Commissioning of Automated Systems, 2008.
[4] Colledani M, Tolio T and Yemane A. Production quality improvement during manufacturing systems ramp-up. CIRP J. Manuf. Sci. Technol., 23:197–206, 2018.
[5] Terwiesch C and Bohn RE. Learning and process improvement during production ramp-up. Int. J. Prod. Econ., 70(1):1–19, 2001.
[6] Klocke F, Stauder J, Mattfeld P and Müller J. Modeling of Manufacturing Technologies During Ramp-up. Procedia CIRP, 51:122–127, 2016.
[7] Doltsinis SC, Ratchev S and Lohse N. A framework for performance measurement during production ramp-up of assembly stations. Eur. J. Oper. Res., 229(1), 85–94, 2013.
[8] Chinnathai MK, Alkan B, Vera D and Harrison R. Pilot to Full-Scale Production: A Battery Module Assembly Case Study. Procedia CIRP,2018, vol. 2018.
[9] Jahangirian M, Eldabi T, Naseer A, Stergioulas LK and Young T. Simulation in manufacturing and business: A review. Eur. J. Oper. Res., 203(1):1–13, 2010.
[10] Vezzetti E, Violante MG and Marcolin F. A benchmarking framework for product lifecycle management (PLM) maturity models. Int. J. Adv. Manuf. Technol., 71(5): 899–918, 2014.
[11] Kuehn W. Digital factory: integration of simulation enhancing the product and production process towards operative control and optimisation. Int. J. Simul., 7(7): 27–39, 2006.
[12] Negahban A and Smith JS. Simulation for manufacturing system design and operation: Literature review and analysis. J. Manuf. Syst., 33(2): 241–261, 2014.
[13] Robinson S. Discrete-event simulation: A primer. Discrete-Event Simulation and System Dynamics for Management Decision Making, John Wiley & Sons, Ltd, 2014, pp. 10–25.
[14] Kampker A, Kreisköther K, Hehl M, Gillen S and Rothe M. Discrete event simulation approach considering scalable systems and non-expert users in the early phase of production planning for electric powertrains. International Conference on Industrial Technology and Management (ICITM), 2017, pp. 139–144.
[15] Ghani U, Monfared R and Harrison R. Integration approach to virtual-driven discrete event simulation for manufacturing systems. Int. J. Comput. Integr. Manuf., 28(8): 844–860, 2015.
[16] Neumeyer S, Exner K, Kind S, Hayka H and Stark R. Virtual Prototyping and Validation of Cpps within a New Software Framework. 2017.
[17] Soto-acosta P, Placer-maruri E and Perez-gonzalez D. A case analysis of a product lifecycle information management framework for SMEs. Int. J. Inf. Manage., 36: 240–244, 2016.
[18] Zhang J and Agyapong-Kodua K. Integrated Ontologies in Support of Factory Systems Design. IFAC-PapersOnLine, 48(3): 2095–2102, 2015.
[19] Lohse N. Towards an ontology framework for the integrated design of modular assembly systems. no. May. 2006.
[20] Chinnathai MK, Alkan B and Harrison R. Convertibility Evaluation of Automated Assembly System Designs for High Variety Production. Procedia CIRP, 2017, vol. 60.
[21] Lumsden J, Hall H and Cruickshank P. Ontology definition and construction , and epistemological adequacy for systems interoperability : A practitioner analysis. 2011.
[22] Musen MA. The protégé project: a look back and a look forward. {AI} Matters, 1(4): 4–12, 2015.

Bibliography

Aggogeri, F., Faglia, R., Mazzola, M. and Merlo, A. [2015], 'Automating the simulation of SME processes through a discrete event parametric model', *International Journal of Engineering Business Management* **7**, 7–4.

Ahmad, M. [2017], An ontology-based approach for integrating engineering workflows for industrial assembly automation systems, PhD thesis, University of Warwick.

Al-Najjar, B. [1996], 'Total quality maintenance', *Journal of Quality in Maintenance Engineering* .

Alkan, B. [2018], A complexity modelling approach to support early life-cycle phases of assembly automation systems, PhD thesis, University of Warwick.

Almgren, H. [2000], 'Pilot production and manufacturing start-up: the case of volvo S80', *International Journal of Production Research* **38**(17), 4577–4588.

Altair [April 2021], 'The pitfalls of a "trial and error" approach to manufacturing'. **URL:** *https://www.altair.com/newsroom/articles/the-pitfalls-of-a-trial-and-error-approach-to-manufacturing/*

Azab, A., AlGeddawy, T. et al. [2012], 'Simulation methods for changeable manufacturing', *Procedia CIRP* **3**, 179–184.

Baines, T. S., Asch, R., Hadfield, L., Mason, J., Fletcher, S. and Kay, J. M. [2005], 'Towards a theoretical framework for human performance modelling within manufacturing systems design', *Simulation modelling practice and theory* **13**(6), 486–504.

Bangsow, S. [2012], 'Use cases of discrete event simulation', *Appliance and research, Berlin, New York* .

Banks, J. [1998], *Handbook of simulation: principles, methodology, advances, applications, and practice*, John Wiley & Sons.

Barlas, P., Dagkakis, G., Heavey, C., Gaffney, B., Young, P. and Geraghty, J. [2014], 'Test implementation and initialisation of a simulation model using CMSD', *Procedia CIRP* **25**, 276–282.

Bathelt, J., Jönsson, A., Bacs, C., Dierssen, S., Meier, M. et al. [2005], Applying the new VDI design guideline 2206 on mechatronic systems controlled by a PLC, *in* 'DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design, Melbourne, Australia, 15.-18.08. 2005', pp. 415–416.

Bensmaine, A., Dahane, M. and Benyoucef, L. [2013], 'A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment', *Computers & Industrial Engineering* **66**(3), 519–524.

Bergström, P. and Dunford, J. [2007], 'Eurostep and John Stark Associates Technology White Paper'.

Betti, F., Boer, E. d. and Giraud, Y. [2020], Industry's fast mover advantage: Enterprise value from digital factories, Technical report, McKinsey Company.

Bishop, C. [2015], System engineering fundamentals, *in* 'Modeling and Simulation in the Systems Engineering Life Cycle', Springer, pp. 257–271.

Bloomfield, R., Mazhari, E., Hawkins, J. and Son, Y.-J. [2012], 'Interoperability of manufacturing applications using the core manufacturing simulation data (CMSD) standard information model', *Computers & Industrial Engineering* **62**(4), 1065–1079.

Bottani, E. and Vignali, G. [2019], 'Augmented reality technology in the manufacturing industry: A review of the last decade', *IISE Transactions* **51**(3), 284–310.

Boucelma, O., Castano, S., Goble, C., Josifovski, V., Lacroix, Z. and Ludäscher, B. [2002], 'Report on the EDBT'02 panel on scientific data integration', *ACM SIGMOD Record* **31**(4), 107–112.

Brailsford, S. [2014], 'Theoretical comparison of discrete-event simulation and system dynamics', *Discrete-event simulation and system dynamics for management decision making* pp. 105–124.

Brailsford, S., Churilov, L., Dangerfield, B. et al. [2014], *Discrete-event simulation and system dynamics for management decision making*, Wiley Online Library.

Branke, J., Branke, J., Deb, K., Miettinen, K. and Słowiński, R. [2008], *Multiobjective optimization: Interactive and evolutionary approaches*, Vol. 5252, Springer Science & Business Media.

Brown, R. G. and Caddick, J. M. [2003], The factory is virtual... the savings are real, Technical report, DELMIA CORP TROY MI.

BSI [2016], Industrial-process measurement, control and automation. Digital factory framework. General principles, Standard.

Bull, M., Chavali, R. and Mascarin, A. [2008], 'Benefit analysis: Use of aluminum structures in conjunction with alternative powertrain technologies in automobiles', *IBIS Associated Inc* .

Burt, T., G. J. [June 19, 2002], Chrysler seeks savings through efficiency, Technical report, Financial Times.

Butter, M., de Heide, M., Montalvo, C., Dittrich, K., Seiffert, L., Navazo, L., Thielmann, A., Braun, A., Meister, M., Holden, D., Livesey, F., O'Sullivan, E., Hartmann, C., Zaldua, M., Oliveri, N. and Turno, L. [2015], Pilot production in key enabling technologies: Crossing the valley of death and boosting the industrial deployment of key enabling technologies in europe, Technical report.

Caggiano, A. [2010], 'Digital factory concept implementation for flexible and reconfigurable manufacturing systems modelling and analysis', *Universita Degli Studi Di Napoli Federico II, Naples* .

Caggiano, A., Caiazzo, F. and Teti, R. [2015], 'Digital factory approach for flexible and efficient manufacturing systems in the aerospace industry', *Procedia CIRP* **37**, 122–127.

Caggiano, A. and Teti, R. [2012], ‘Digital manufacturing cell design for performance increase’, *Procedia CIRP* **2**, 64–69.

Caggiano, A. and Teti, R. [2018], ‘Digital factory technologies for robotic automation and enhanced manufacturing cell design’, *Cogent Engineering* **5**(1), 1426676.

Camba, J. D., Contero, M., Company, P. and Pérez, D. [2017], ‘On the integration of model-based feature information in product lifecycle management systems’, *International journal of information management* **37**(6), 611–621.

Carrillo, J. E. and Franza, R. M. [2006], ‘Investing in product development and production capabilities: The crucial linkage between time-to-market and ramp-up time’, *European Journal of Operational Research* **171**(2), 536–556.
URL: *https://www.sciencedirect.com/science/article/pii/S037722170400623X*

Chase, R. B. and Aquilano, N. J. [1977], *Production and operations management: A life cycle approach*, McGraw-Hill/Irwin.

Chemicals, H. S. [2018], Driving lab performance and productivity, Technical report, Germany.

Chinnathai, M. K., Al-Mowafy, Z., Alkan, B., Vera, D. and Harrison, R. [2019], ‘A framework for pilot line scale-up using digital manufacturing’, *Procedia CIRP* **81**, 962–967.

Chinnathai, M. K., Alkan, B. and Harrison, R. [2017], ‘Convertibility evaluation of automated assembly system designs for high variety production’, *Procedia CIRP* **60**, 74–79.

Chinnathai, M. K., Alkan, B., Vera, D. and Harrison, R. [2018], ‘Pilot to full-scale production: A battery module assembly case study’, *Procedia CIRP* **72**, 796–801.

Chinnathai, M. K., Günther, T., Ahmad, M., Stocker, C., Richter, L., Schreiner, D., Vera, D., Reinhart, G. and Harrison, R. [2017], ‘An application of physical flexibility and software reconfigurability for the automation of battery module assembly’, *Procedia CIRP* **63**, 604–609.

Choi, S. S., Yoon, T. H. and Noh, S. D. [2010], 'XML-based neutral file and PLM integrator for PPR information exchange between heterogeneous PLM systems', *International Journal of Computer Integrated Manufacturing* **23**(3), 216–228.

Chryssolouris, G. [2013], *Manufacturing systems: theory and practice*, Springer Science & Business Media.

Chryssolouris, G., Mavrikios, D., Papakostas, N., Mourtzis, D., Michalos, G. and Georgoulias, K. [2009], 'Digital manufacturing: history, perspectives, and outlook', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **223**(5), 451–462.

Chung, C. A. [2003], *Simulation modeling handbook: a practical approach*, CRC press.

CMSD, P. D. G. [2010], SISO-STD-008-2010: Standard for core manufacturing simulation data (CMSD) - UML model, Standard, Simulation Interoperability Standards Organisation, Orlando,USA.

CMSD, P. D. G. [2012], SISO-STD-008-01-2012: Standard for core manufacturing simulation data (CMSD) - XML, Standard, Simulation Interoperability Standards Organisation, Orlando,USA.

Colledani, M., Tolio, T. and Yemane, A. [2018], 'Production quality improvement during manufacturing systems ramp-up', *CIRP Journal of Manufacturing Science and Technology* **23**, 197–206.

Deb, K., Mohan, M. and Mishra, S. [2003], Towards a quick computation of well-spread pareto-optimal solutions, *in* 'International Conference on Evolutionary Multi-Criterion Optimization', Springer, pp. 222–236.

Deif, A. M. and ElMaraghy, H. A. [2007*a*], 'Assessing capacity scalability policies in RMS using system dynamics', *International journal of flexible manufacturing systems* **19**(3), 128–150.

Deif, A. M. and ElMaraghy, W. [2007*b*], 'Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system', *The International Journal of Advanced Manufacturing Technology* **32**(5-6), 557–562.

Devlin, B., Gray, J., Laing, B. and Spix, G. [1999], ‘Scalability terminology: Farms, clones, partitions and packs: RACS and RAPS’.

Doltsinis, S. C., Ratchev, S. and Lohse, N. [2013], ‘A framework for performance measurement during production ramp-up of assembly stations’, *European journal of operational research* **229**(1), 85–94.

Drath, R., Luder, A., Peschke, J. and Hundt, L. [2008], AutomationML-the glue for seamless automation engineering, *in* ‘2008 IEEE International Conference on Emerging Technologies and Factory Automation’, IEEE, pp. 616–623.

Economist, T. [n.d.], ‘The growth of lithium-ion battery power’.
URL: *https://www.economist.com/graphic-detail/2017/08/14/the-growth-of-lithium-ion-battery-power*

ElMaraghy, H. A. [2005], ‘Flexible and reconfigurable manufacturing systems paradigms’, *International journal of flexible manufacturing systems* **17**(4), 261–276.

Elstner, S. and Krause, D. [2014], ‘Methodical approach for consideration of ramp-up risks in the product development of complex products’, *Procedia CIRP* **20**, 20–25.

ESPRIT Consortium AMICE Staff, C. [1993], *Open System Architecture for CIM*, Springer-Verlag.

Falkman, P., Westman, F. and Modig, C. [2009], Verification of operation sequences in process simulate by connecting a formal verification tool, *in* ‘2009 IEEE International Conference on Control and Automation’, IEEE, pp. 1207–1212.

Faure, A., York, P. and Rowe, R. [2001], ‘Process control and scale-up of pharmaceutical wet granulation processes: a review’, *European Journal of Pharmaceutics and Biopharmaceutics* **52**(3), 269–277.

Ferrer, B. R., Ahmad, B., Lobov, A., Vera, D., Lastra, J. L. M. and Harrison, R. [2015], A knowledge-based solution for automatic mapping in component based automation systems, *in* ‘2015 IEEE 13th international conference on industrial informatics (indin)’, IEEE, pp. 262–268.

Ferrer, B. R., Ahmad, B., Vera, D., Lobov, A., Harrison, R. and Lastra, J. L. M. [2016], 'Product, process and resource model coupling for knowledge-driven assembly automation', *at-Automatisierungstechnik* **64**(3), 231–243.

Fishman, G. S. [2013], *Discrete-event simulation: modeling, programming, and analysis*, Springer Science & Business Media.

Ford, H. and Crowther, S. [1922], *My life and work*, CreateSpace Independent Publishing Platform.

Fricke, E. and Schulz, A. P. [2005], 'Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle', *Systems Engineering* **8**(4), no–no.

Garms, F., Jansen, C., Schmitz, C., Hallerstede, S. and Tschiesner, A. [2019], Capturing value at scale in discrete manufacturing with industry 4.0, Technical report, McKinsey Company.

Ghani, U. [2013], System optimisation through the integration of virtual engineering and discrete event simulation, PhD thesis, Loughborough University.

Ghani, U., Monfared, R. and Harrison, R. [2015], 'Integration approach to virtual-driven discrete event simulation for manufacturing systems', *International Journal of Computer Integrated Manufacturing* **28**(8), 844–860.

Gilmore, J. H., Pine, B. J. et al. [1997], 'The four faces of mass customization', *Harvard business review* **75**(1), 91–102.

Gjerdrum, J., Shah, N. and Papageorgiou, L. G. [2001], 'A combined optimization and agent-based approach to supply chain modelling and performance assessment', *Production Planning & Control* **12**(1), 81–88.

Goldberg, D. E. [1989], *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., USA.

Gomez, A. L. and Strathy, W. A. [2001], Engineering aspects of process scale-up and pilot plant design, *in* 'Pharmaceutical Process Scale-Up', CRC Press, pp. 333–346.

Gosavi, A. [2015], *Simulation-based optimization: Parametric Optimization Techniques and Reinforcement Learning*, 2nd edn, Springer.

Goto, S. and Yoshie, O. [2019], *PLM Case Studies in Japan*, Springer International Publishing, Cham, pp. 117–124.
URL: *https://doi.org/10.1007/978-3-030-16134-7_10*

Gruber, T. R. et al. [1993], 'A translation approach to portable ontology specifications', *Knowledge acquisition* **5**(2), 199–221.

Guidebook on Design, Construction and Operation of Pilot Plants for Uranium Ore Processing [1990], number 314 *in* 'Technical Reports Series', International atomic energy agency, Vienna.

Guschinskaya, O., Dolgui, A., Guschinsky, N. and Levin, G. [2008], 'A heuristic multi-start decomposition approach for optimal design of serial machining lines', *European Journal of Operational Research* **189**(3), 902–913.

Haller, M., Peikert, A. and Thoma, J. [2003], 'Cycle time management during production ramp-up', *Robotics and Computer-Integrated Manufacturing* **19**(1-2), 183–188.

Harding, J. and Yu, B. [1999], 'Information-centred enterprise design supported by a factory data model and data warehousing', *Computers in Industry* **40**(1), 23–36.

Harjunkoski, I. and Bauer, R. [2014], 'Sharing data for production scheduling using the ISA-95 standard', *Frontiers in Energy Research* **2**, 44.

He, D., Lobov, A. and Lastra, J. M. [2012], ISA-95 tool for enterprise modeling, *in* 'Proceeding of ICONS 2012, The Seventh International Conference on Systems', pp. 83–87.

Heilala, J., Vatanen, S., Tonteri, H., Montonen, J., Lind, S., Johansson, B. and Stahre, J. [2008], Simulation-based sustainable manufacturing system design, *in* '2008 Winter simulation conference', IEEE, pp. 1922–1930.

Hewett, A. [2010], Product lifecycle management (PLM): Critical issues and challenges in implementation, *in* 'Information Technology and Product Development', Springer, pp. 81–105.

Hitomi, K. [1996], 'Manufacturing excellence for 21st century production', *Technovation* **16**(1), 33 – 41.
URL: *http://www.sciencedirect.com/science/article/pii/0166497295000186*

Hitomi, K. [2017], *Manufacturing systems engineering: a unified approach to manufacturing technology, production management and industrial economics*, Routledge.

Holweg, M. [2007], 'The genealogy of lean production', *Journal of operations management* **25**(2), 420–437.

Hu, S. J. [2013], 'Evolving paradigms of manufacturing: from mass production to mass customization and personalization', *Procedia Cirp* **7**, 3–8.

Hu, S. J., Ko, J., Weyand, L., ElMaraghy, H. A., Lien, T. K., Koren, Y., Bley, H., Chryssolouris, G., Nasr, N. and Shpitalni, M. [2011], 'Assembly system design and operations for product variety', *CIRP annals* **60**(2), 715–733.

Hundt, L., Drath, R., Lüder, A. and Peschke, J. [2008], Seamless automation engineering with AutomationML®, *in* '2008 IEEE International Technology Management Conference (ICE)', IEEE, pp. 1–8.

INCOSE, T. [2007], 'Systems engineering vision 2020', *INCOSE, San Diego, CA* **26**, 2019.

Integrity, P. S. D. [2012], 'ENOVIA collaborative design for Pro/ENGINEER'.

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K. and Young, T. [2010], 'Simulation in manufacturing and business: A review', *European Journal of Operational Research* **203**(1), 1–13.

Jain, S., Choong, N. F., Aye, K. M. and Luo, M. [2001], 'Virtual factory: an integrated approach to manufacturing systems modeling', *International Journal of Operations & Production Management* .

Jain, V. and Singh, M. [2013], 'Ontology development and query retrieval using protégé tool', *International Journal of Intelligent Systems and Applications (IJISA)* **5**(9), 67–75.

Jiang, S., Ong, S. and Nee, A. [2014], 'An AR-based hybrid approach for facility layout planning and evaluation for existing shop floors', *The International Journal of Advanced Manufacturing Technology* **72**(1-4), 457–473.

Kádár, B., Pfeiffer, A. and Monostori, L. [2004], Discrete event simulation for supporting production planning and scheduling decisions in digital factories, *in* 'Proceedings of the 37th CIRP international seminar on manufacturing systems', Citeseer, pp. 444–448.

Kampker, A., Kreisköther, K., Hehl, M., Gillen, S. and Rothe, M. [2017], Discrete event simulation approach considering scalable systems and non-expert users in the early phase of production planning for electric powertrains, *in* '6th International Conference on Industrial Technology and Management (ICITM)', IEEE, pp. 139–144.

Kans, M. and Ingwald, A. [2008], 'Common database for cost-effective improvement of maintenance performance', *International journal of production economics* **113**(2), 734–747.

Karuppan, C. M. and Kepes, S. [2006], 'The strategic pursuit of mix flexibility through operators' involvement in decision making', *International Journal of Operations & Production Management* .

Kibira, D. and McLean, C. [2002], Virtual reality simulation of a mechanical assembly production line, *in* 'Proceedings of the Winter Simulation Conference', Vol. 2, IEEE, pp. 1130–1137.

Kinaxis [2018], An integrated approach to global capacity management for automotive manufacturers, Technical report, Canada.

Kinzoku, M. [2008], Electrodeposited copper foil plant in malaysia to significantly enhance production capacity, Technical report, Malaysia.

Kinzoku, M. [2018*a*], Increase in production capacity of ultra-fine copper powder for MLCC, Technical report, Japan.

Kinzoku, M. [2018*b*], Mitsui kinzoku completes increase of production capacity for ultra-thin electrodeposited copper foil with carrier microthintm, Technical report, Malaysia.

Kinzoku, M. [2019], Mitsui kinzoku completes increase in production capacity of VSP® electro-deposited copper foil for high-frequency circuit boards, Technical report, Taiwan.

Klocke, F., Stauder, J., Mattfeld, P. and Müller, J. [2016], 'Modeling of manufacturing technologies during ramp-up', *Procedia CIRP* **51**, 122–127.

Knublauch, H., Fergerson, R. W., Noy, N. F. and Musen, M. A. [2004], The protégé OWL plugin: An open development environment for semantic web applications, *in* 'International semantic web conference', Springer, pp. 229–243.

Konak, A., Coit, D. W. and Smith, A. E. [2006], 'Multi-objective optimization using genetic algorithms: A tutorial', *Reliability Engineering & System Safety* **91**(9), 992–1007.

Koren, Y. [2006], General RMS characteristics. comparison with dedicated and flexible systems, *in* 'Reconfigurable manufacturing systems and transformable factories', Springer, pp. 27–45.

Koren, Y. [2010], *The global manufacturing revolution: product-process-business integration and reconfigurable systems*, Vol. 80, John Wiley & Sons.

Koren, Y. and Ulsoy, A. G. [2002], 'Reconfigurable manufacturing system having a production capacity method for designing same and method for changing its production capacity'. US Patent 6,349,237.

Koren, Y., Wang, W. and Gu, X. [2017], 'Value creation through design for scalability of reconfigurable manufacturing systems', *International Journal of Production Research* **55**(5), 1227–1242.

Kuhn, W. [2006], 'Digital factory–integration of simulation enhancing the product and production process towards operative control and optimisation', *International Journal of Simulation* **7**(7), 27–39.

KUKA [2020], 'Flexibility in battery production'.
URL: *https://www.kuka.com/en-gb/industries/solutions-database/2020/04/battery-module-line-elringklinger*

Kurawarwala, A. A. and Matsuo, H. [1996], 'Forecasting and inventory management of short life-cycle products', *Operations Research* **44**(1), 131–150.

Lanza, G. and Sauer, A. [2012], 'Simulation of personnel requirements during production ramp-up', *Production Engineering* **6**(4-5), 395–402.

Law, A. M. [1986], 'Introduction to simulation: A powerful tool for analyzing complex manufacturing systems', *Industrial Engineering* **18**(5), 46–63.

Lee, C. G. and Park, S. C. [2014], 'Survey on the virtual commissioning of manufacturing systems', *Journal of Computational Design and Engineering* **1**(3), 213–222.

Lee, S., Harrison, R. and West, A. [2005], 'A component-based control system for agile manufacturing', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **219**(1), 123–135.

Lee, S., Harrison, R., West, A. and Ong, M. [2007], 'A component-based approach to the design and implementation of assembly automation system', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **221**(5), 763–773.

Leuenberger, H. [2001], 'New trends in the production of pharmaceutical granules: the classical batch concept and the problem of scale-up', *European Journal of Pharmaceutics and Biopharmaceutics* **52**(3), 279–288.

Levin, M., ed. [2001], *Pharmaceutical process scale-up*, CRC Press.

Liu, Z., Magnus, S., Krause, J. and Diedrich, C. [2014], Concept for modelling and testing of individual mechatronic components for manufacturing plant simulation, *in* 'Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)', IEEE, pp. 1–7.

Liu, Z., Suchold, N. and Diedrich, C. [2012], Virtual commissioning of automated systems, *in* 'Automation', IntechOpen.

Lohse, N. [2006], Towards an ontology framework for the integrated design of modular assembly systems, PhD thesis, University of Nottingham Nottingham.

Lohse, N., Hirani, H. and Ratchev, S. [2005], 'Equipment ontology for modular reconfigurable assembly systems', *International Journal of Flexible Manufacturing Systems* **17**(4), 301.

Macal, C. M. and North, M. J. [2010], Toward teaching agent-based simulation, *in* 'Proceedings of the 2010 Winter Simulation Conference', IEEE, pp. 268–277.

Mahdavi, I., Shirazi, B. and Solimanpur, M. [2010], 'Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems', *Simulation Modelling Practice and Theory* **18**(6), 768–786.

Maidstone, R. [2012], 'Discrete event simulation, system dynamics and agent based simulation: Discussion and comparison', *System* **1**(6), 1–6.

Makris, S., Michalos, G. and Chryssolouris, G. [2012], 'Virtual commissioning of an assembly cell with cooperating robots', *Adv. Decis. Sci.* **2012**, 428060:1–428060:11.

Manzini, M., Unglert, J., Gyulai, D., Colledani, M., Jauregui-Becker, J. M., Monostori, L. and Urgo, M. [2018], 'An integrated framework for design, management and operation of reconfigurable assembly systems', *Omega* **78**, 69–84.

Maropoulos, P. G. and Ceglarek, D. [2010], 'Design verification and validation in product lifecycle', *CIRP annals* **59**(2), 740–759.

Mas, F., Arista, R., Oliva, M., Hiebert, B., Gilkerson, I. and Rios, J. [2015], 'A review of PLM impact on US and EU aerospace industry', *Procedia Engineering* **132**, 1053–1060.

MathWorks [2021*a*], 'Opc toolbox: User's guide (r2021a)'.
URL: *https://uk.mathworks.com/help/pdf$_d$oc/opc/opc.pdf*

MathWorks [2021*b*], 'Optimization toolbox: User's guide (r2021a)'.
URL: *https://uk.mathworks.com/help/pdf$_d$oc/optim/optim.pdf*

McLean, C. and Leong, S. [2001], The role of simulation in strategic manufacturing, *in* 'Proceedings of the 33nd Conference on Winter Simulation', Citeseer, pp. 1478–1486.

Mcnulty, T. [1998], 'Innovative technology; its development and commercialization', *Managing Innovation in the Minerals Industry, Society for Mining, Metallurgy, and Exploration, Inc., Littleton, Colorado* .

Michalos, G., Fysikopoulos, A., Makris, S., Mourtzis, D. and Chryssolouris, G. [2015], 'Multi criteria assembly line design and configuration–an automotive case study', *CIRP Journal of Manufacturing Science and Technology* **9**, 69–87.

Mieth, C., Meyer, A. and Henke, M. [2019], 'Framework for the usage of data from real-time indoor localization systems to derive inputs for manufacturing simulation', *Procedia CIRP* **81**, 868–873.

Mirdamadi, S., Fontanili, F. and Dupont, L. [2007], Discrete event simulation-based real-time shop floor control, *in* 'Proc. 21st Eur. Conf. Modelling and Simulation (ECMS 2007)', pp. 235–240.

Monden, Y. [2011], *Toyota production system: An integrated approach to just-in-time*, Springer.

Mourtzis, D. [2020*a*], 'Simulation in the design and operation of manufacturing systems: state of the art and new trends', *International Journal of Production Research* **58**(7), 1927–1949.

Mourtzis, D. [2020*b*], 'Simulation in the design and operation of manufacturing systems: state of the art and new trends', *International Journal of Production Research* **58**(7), 1927–1949.
URL: *https://doi.org/10.1080/00207543.2019.1636321*

Mourtzis, D. and Doukas, M. [2012], 'A web-based platform for customer integration in the decentralised manufacturing of personalised products', *Procedia CIRP* **3**, 209–214.

Mourtzis, D., Doukas, M. and Bernidaki, D. [2014], 'Simulation in manufacturing: Review and challenges', *Procedia Cirp* **25**, 213–229.

Mourtzis, D., Doukas, M., Psarommatis, F., Giannoulis, C. and Michalos, G. [2014], 'A web-based platform for mass customisation and personalisation', *CIRP Journal of Manufacturing Science and Technology* **7**(2), 112–128.

Müller, E. and Horbach, S. [2012], Building blocks in an experimental and digital factory, *in* 'Enabling Manufacturing Competitiveness and Economic Sustainability', Springer, pp. 592–597.

Mus'ab H, A. [2017], A Proactive Framework within a Virtual Engineering Environment for Assembly System Energy Optimisation, PhD thesis, University of Warwick.

Nahm, J. and Steinfeld, E. S. [2014], 'Scale-up nation: China's specialization in innovative manufacturing', *World Development* **54**, 288–300.

Nazzal, D., Mollaghasemi, M. and Anderson, D. [2006], 'A simulation-based evaluation of the cost of cycle time reduction in agere systems wafer fabrication facility—a case study', *International Journal of Production Economics* **100**(2), 300–313.

Negahban, A. and Smith, J. S. [2014], 'Simulation for manufacturing system design and operation: Literature review and analysis', *Journal of Manufacturing Systems* **33**(2), 241–261.

Oates, C. [2000], 'The use of focus groups in social science research', *Research training for social scientists* pp. 186–195.

O'Connor, M. J. and Das, A. K. [2009], SQWRL: a query language for OWL, *in* 'OWLED', Vol. 529.

Ohno, T. [1988], *Toyota production system: beyond large-scale production*, crc Press.

Okoh, C., Roy, R. and Mehnen, J. [2017], 'Predictive maintenance modelling for through-life engineering services', *Procedia CIRP* **59**, 196–201.

Patel, M., Saunders, J., Sergeant, N. and Tilia, W. [2016], Ramping up at warp speed, Technical report, McKinsey Company.

Penciuc, D., Durupt, A., Belkadi, F., Eynard, B. and Rowson, H. [2014], 'Towards a PLM interoperability for a collaborative design support system', *Procedia Cirp* **25**, 369–376.

Popplewell, K., Harding, J., Ricardo, C. and Poler, R., eds [2010], *Enterprise Interoperability IV*, 1 edn, Springer.

Prajapat, N. and Tiwari, A. [2017], 'A review of assembly optimisation applications using discrete event simulation', *International Journal of Computer Integrated Manufacturing* **30**(2-3), 215–228.

Putnik, G., Sluga, A., ElMaraghy, H., Teti, R., Koren, Y., Tolio, T. and Hon, B. [2013], 'Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap', *CIRP Annals* **62**(2), 751–774.

Reinhardt, H., Weber, M. and Putz, M. [2019], 'A survey on automatic model generation for material flow simulation in discrete manufacturing', *Procedia CIRP* **81**, 121–126.

Remiel, F. M., Cauvin, A. and Ferrarini, A. [2014], 'Conceptual design and simulation of an automotive body shop assembly line', *IFAC Proceedings Volumes* **47**(3), 760–765.

Robinson, S. [2007], 'A statistical process control approach to selecting a warm-up period for a discrete-event simulation', *European Journal of Operational Research* **176**(1), 332–346.
URL: *https://www.sciencedirect.com/science/article/pii/S0377221705005643*

Robinson, S. and Brailsford, S. [2014], 'Discrete-event simulation: A primer', *Discrete-event simulation and system dynamics for management decision making* pp. 10–25.

Rohmer, E., Singh, S. P. and Freese, M. [2013], V-REP: A versatile and scalable robot simulation framework, *in* '2013 IEEE/RSJ International Conference on Intelligent Robots and Systems', IEEE, pp. 1321–1326.

Ross, A. M., Rhodes, D. H. and Hastings, D. E. [2008], 'Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value', *Systems Engineering* **11**(3), 246–262.

Saariluoma, H., Piiroinen, A., Unt, A., Hakanen, J., Rautava, T. and Salminen, A. [2020], 'Overview of optical digital measuring challenges and technologies

in laser welded components in ev battery module design and manufacturing', *Batteries* **6**(3).
URL: *https://www.mdpi.com/2313-0105/6/3/47*

Sacco, M., Pedrazzoli, P. and Terkaj, W. [2010], VFF: virtual factory framework, *in* '2010 IEEE International Technology Management Conference (ICE)', IEEE, pp. 1–8.

Segonds, F., Cohen, G., Veron, P. and Peyceré, J. [2016], 'PLM and early stages collaboration in interactive design, a case study in the glass industry', *International Journal on Interactive Design and Manufacturing (IJIDeM)* **10**(2), 95–104.

Seidel, S., Donath, U. and Haufe, J. [2012], Towards an integrated simulation and virtual commissioning environment for controls of material handling systems, *in* 'Proceedings of the 2012 Winter Simulation Conference (WSC)', pp. 1–12.

Seleim, Azab, A. and AlGeddawy, T. [2012], 'Simulation methods for changeable manufacturing', *Procedia CIRP* **3**, 179 – 184. 45th CIRP Conference on Manufacturing Systems 2012.
URL: *http://www.sciencedirect.com/science/article/pii/S2212827112002041*

Sendler, U. [2009], Siemens PLM software, *in* 'Das PLM-Kompendium', Springer, pp. 185–195.

Shao, G., Bengtsson, N. and Johansson, B. [2010], Interoperability for simulation of sustainable manufacturing, *in* 'Proceedings of the 2010 Spring Simulation Multiconference', pp. 1–8.

Shibasaki, M., Warburg, N. and Eyerer, P. [2006], Upscaling effect and life cycle assessment, *in* 'Proceedings of the 13th CIRP International Conference on Life Cycle Engineering, Leuven, Belgium', Vol. 31.

Siemens [2009], 'Siemens PLM Software's Teamcenter', *Aircraft Engineering and Aerospace Technology* **81**(2).

Singh, S. and Misra, S. C. [2019], *Significance of Cloud PLM in Industry 4.0*, Springer International Publishing, Cham, pp. 249–255.
URL: *https://doi.org/10.1007/978-3-030-16134-7_19*

Slack, N. [1987], ‘The flexibility of manufacturing systems’, *International Journal of Operations & Production Management* .

Slamanig, M. and Winkler, H. [2011], ‘An exploration of ramp-up strategies in the area of mass customisation’, *International Journal of Mass Customisation* **4**(1-2), 22–43.

Soto-Acosta, P., Placer-Maruri, E. and Perez-Gonzalez, D. [2016], ‘A case analysis of a product lifecycle information management framework for SMEs’, *International Journal of Information Management* **36**(2), 240–244.

Stanton, N. A., Hedge, A., Brookhuis, K., Salas, E. and Hendrick, H. W. [2004], *Handbook of human factors and ergonomics methods*, CRC press.

Stauder, J., Buchholz, S., Klocke, F. and Mattfeld, P. [2014], ‘A new framework to evaluate the process capability of production technologies during production ramp-up’, *Procedia CIRP* **20**, 126–131.

Surbier, L., Alpan, G. and Blanco, E. [2014], ‘A comparative study on production ramp-up: state-of-the-art and new challenges’, *Production Planning & Control* **25**(15), 1264–1286.

Tako, A. A. and Robinson, S. [2012], ‘The application of discrete event simulation and system dynamics in the logistics and supply chain context’, *Decision support systems* **52**(4), 802–815.

Terkaj, W., Pedrielli, G. and Sacco, M. [2012], Virtual factory data model, *in* ‘Proceedings of the workshop on ontology and semantic web for manufacturing, Graz, Austria’, pp. 29–43.

Terkaj, W. and Urgo, M. [2015], ‘A virtual factory data model as a support tool for the simulation of manufacturing systems’, *Procedia CIRP* **28**, 137–142.

Terwiesch, C., Bohn, R. and Chea, K. [2001], ‘International product transfer and production ramp-up: a case study from the data storage industry’, *R&D Management* **31**(4), 435–451.

Terwiesch, C. and Bohn, R. E. [2001], ‘Learning and process improvement during production ramp-up’, *International journal of production economics* **70**(1), 1–19.

Tocher, K. and Owen, D. [2008], 'The automatic programming of simulations', *Journal of Simulation* **2**(3), 143–152.

Tolio, T., Sacco, M., Terkaj, W. and Urgo, M. [2013], 'Virtual factory: An integrated framework for manufacturing systems design and analysis', *Procedia CIRP* **7**, 25–30.

Tolk, A. and Muguira, J. A. [2003], The levels of conceptual interoperability model, *in* 'Proceedings of the 2003 fall simulation interoperability workshop', Vol. 7, Citeseer, pp. 1–11.

Tracht, K. and Hogreve, S. [2012], Decision making during design and reconfiguration of modular assembly lines, *in* 'Enabling Manufacturing Competitiveness and Economic Sustainability', Springer, pp. 105–110.

Trstenjak, M. and Cosic, P. [2017], 'Process planning in industry 4.0 environment', *Procedia Manufacturing* **11**, 1744–1750.

Tsinontides, S., Rajniak, P., Pham, D., Hunke, W., Placek, J. and Reynolds, S. [2004], 'Freeze drying—principles and practice for successful scale-up to manufacturing', *International journal of pharmaceutics* **280**(1-2), 1–16.

Up, H. S. [2017], 'A sustainable pathway for the global energy transition', *Hydrogen Council, November* .

Uschold, M. and Gruninger, M. [1996], 'Ontologies: Principles, methods and applications', *The knowledge engineering review* **11**(2), 93–136.

Venkataraman, K., Ramnath, B. V., Kumar, V. M. and Elanchezhian, C. [2014], 'Application of value stream mapping for reduction of cycle time in a machining process', *Procedia Materials Science* **6**, 1187–1196.

Vezzetti, E., Violante, M. G. and Marcolin, F. [2014], 'A benchmarking framework for product lifecycle management (PLM) maturity models', *The International Journal of Advanced Manufacturing Technology* **71**(5-8), 899–918.

Volkswagen [June 2019], 'Volkswagen and northvolt form joint venture for battery production'.

URL: *https://www.volkswagen-newsroom.com/en/press-releases/volkswagen-and-northvolt-form-joint-venture-for-battery-production-5316*

Volvo, A. [2007], Volvo group annual report, Technical report, Sweden.

Wang, G. G. [2002], 'Definition and review of virtual prototyping', *J. Comput. Inf. Sci. Eng.* **2**(3), 232–236.

Wang, L., Ng, A. H. and Deb, K. [2011], *Multi-objective evolutionary optimisation for product design and manufacturing*, Springer.

Wang, W. and Koren, Y. [2012], 'Scalability planning for reconfigurable manufacturing systems', *Journal of manufacturing systems* **31**(2), 83–91.

Westkämper, E. [2007], 'Strategic development of factories under the influence of emergent technologies', *CIRP annals* **56**(1), 419–422.

Wiendahl, H.-P. and Scholtissek, P. [1994], 'Management and control of complexity in manufacturing', *CIRP annals* **43**(2), 533–540.

Williams, K., Haslam, C. and Williams, J. [1992], 'Ford versus 'fordism': The beginning of mass production?', *Work, Employment and Society* **6**(4), 517–555.
URL: *https://doi.org/10.1177/095001709264001*

Wirges, M., Funke, A., Serno, P., Knop, K. and Kleinebudde, P. [2013], 'Development and in-line validation of a process analytical technology to facilitate the scale up of coating processes', *Journal of pharmaceutical and biomedical analysis* **78**, 57–64.

Wischnewski, R., Rossmann, J. and Stern, O. [2012], New methods to create variants of 3-D simulation models of manufacturing systems, *in* 'Enabling Manufacturing Competitiveness and Economic Sustainability', Springer, pp. 74–79.

Womack, J. P., Jones, D. T. and Roos, D. [2007], *The machine that changed the world: The story of lean production–Toyota's secret weapon in the global car wars that is now revolutionizing world industry*, Simon and Schuster.

Woodward, M., W. B. H. J. [2020], 'Electric vehicles: Setting a course for 2030'.
URL: *https://www2.deloitte.com/uk/en/insights/focus/future-of-mobility/electric-vehicle-trends-2030.html*

Wu, D., Rosen, D. W., Wang, L. and Schaefer, D. [2015], 'Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation', *Computer-Aided Design* **59**, 1–14.

Wu, D., Thames, J. L., Rosen, D. W. and Schaefer, D. [2012], Towards a cloud-based design and manufacturing paradigm: looking backward, looking forward, *in* 'International Design Engineering Technical Conferences and Computers and Information in Engineering Conference', Vol. 45011, American Society of Mechanical Engineers, pp. 315–328.

Xue, J. [2018], *Integration of CAD/CAPP/CAM*, De Gruyter, Berlin, Boston.
URL: *https://www.degruyter.com/view/title/534709*

Zhai, W., Fan, X., Yan, J. and Zhu, P. [2002], 'An integrated simulation method to support virtual factory engineering', *International Journal of CAD/CAM* **2**(1), 39–44.

Zhang, L., Luo, Y., Tao, F., Li, B. H., Ren, L., Zhang, X., Guo, H., Cheng, Y., Hu, A. and Liu, Y. [2014], 'Cloud manufacturing: a new manufacturing paradigm', *Enterprise Information Systems* **8**(2), 167–187.

Zhang, Q. and Cen, S. [2015], *Multiphysics Modeling: Numerical Methods and Engineering Applications: Tsinghua University Press Computational Mechanics Series*, Elsevier.