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.

1. Introduction

1.1. Research background

The current manufacturing era faces a lot of challenges such as increased customisation, complexity and customer demand, shorter product life-cycles and adaptation to new technologies [1]. In particular, the shortened product life-cycles and unpredictable demand variations impose frequent hardware and software modifications to the industrial production lines. However, to stay competitive, industries must be able to rapidly progress from concept/small-scale to operational/full-scale production. To realise this, industries rely on prototype testing, either virtually or physically, to detect and anticipate potential issues that could impact the line in the early stages. This can lead to significant cost and time savings while allowing the industries to stay competitive by shortening the concept to volume duration [2, 3]. The core challenge, however, is that the approach to perform this critical transition is lacking in literature [4].

The term '*production scale-up*' is defined by the authors 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"*. In other words, the scale-up phase involves modifying or transforming the hardware and/or the software of the existing line or redesigning the system to meet the new demand. The performance of the modified line has to be evaluated either by commissioning the line or building virtual models. Therefore, a diverse set of digital tools that are effective in modelling and simulation can be employed to understand the behaviour of manufacturing systems, especially the complex ones, and subsequently predict their performance [2, 5, 6, 7]. Consequently, the adverse effects associated with and imposed by the modifications brought about during scale-up phase can be identified and curtailed before the commencement of the changeover [8, 9].

A detailed survey of white papers and reports reveals that in-

*Malarvizhi Kaniappan Chinnathai: Warwick Manufacturing Group, University of Warwick, Coventry, UK, CV4 7AL; M.Kaniappan-Chinnathai.1@warwick.ac.uk

dustrial practices for scale-up can be classified into automation, improving existing processes, and/or adding more production lines, workstations and factories [10, 11, 12, 13, 14, 15]. Discussions with industrial project partners reveal that the modifications during the changeover phase is carried out as an impromptu procedure that is not reinforced by a robust systematic approach or framework. It is also understood that the industrial scale-up practices heavily rely on personnel experience to modify and improve production lines. However, the changes made to a certain region of the system could inadvertently trigger a disturbance elsewhere; it is difficult to predict such issues based on experience alone. Moreover, such trial and error based scale-up could potentially lead to costly time-consuming sub-optimal solutions that might not provide the desired results.

In the academic body of knowledge, a multitude of discussions on scale-up process and its details have been done in the domains of pharmaceutics and process industries [16, 17, 18, 19, 20]. However, in the manufacturing domain, the focus remains on scale-up management, quantification of scalability and capacity scalability [4, 21, 22, 23, 24, 25]. A significant proportion of the reviewed papers highlight the disturbances and events that occur during the transition from concept to operational phase and the strategies to manage and prevent these disturbances. The existing approaches in industries and academia *i*) generally refer to trial and error based methods that adversely impact the time and cost of performing scale-up projects and *ii*) do not provide a systematic method or workflow to ensure smooth transition during the scale-up process. This further emphasises the need for a framework to support manufacturing system scale-up.

1.2. Research approach

This research proposes a data-driven manufacturing system scale-up decision support that builds upon the methods of kinematic modelling and DES, further enhanced by historical production data and knowledge representation techniques. The methodology is divided into two main stages: *i*) workstation and equipment modelling with kinematic modelling software (Stage one) and *ii*) pilot and production line modelling with DES software (Stage two). In Stage one, a kinematic model of the existing manufacturing system is designed and analysed to understand the product and process specifications. To better support the selection of suitable workstation configurations, the existing kinematic model is coupled with an ontology editor that can query and select equipment that perform the desired assembly process defined by a set of parameters. The selected candidate equipment reflect the potential workstation configurations, the performance of which can further be weighed; the filtered results can be tested and validated with a kinematic modelling software. The corresponding workstation KPIs of each selected solution in Stage one is utilised in Stage two for performing multi-objective simulation optimisation using MATLAB and FlexSim DES software. In simulation optimisation, the values of the decision variables are provided to the DES model and at the end of each simulation run, the system KPIs are passed back to MATLAB for calculating the objective function. Since the DES model is fundamentally constructed using the workstation data obtained from Stage one, the approach provides a novel bottom-up design selection process that ultimately improves the accuracy of the system/assembly line model in DES.

In summary, this research study proposes an overarching framework to support the scale-up of assembly systems through the data integration of two distinct modelling methods, kinematic modelling and DES for workstation level and production line level modelling, respectively. The ultimate goal is to predict and virtually validate the performance and behaviour of the potential assembly system configurations that are represented by the varying quantity, type and arrangement of workstations, equipment, and material handling units to meet the predicted increase in demand.

1.3. Research scope

This study focuses on industrial assembly systems which are decomposed into five levels of granularity, referred to as '*layers*', such as: *i*) component, *ii*) station, *iii*) pilot line, *iv*) production line and *v*) factory. The '*component*' layer represents the highest level of granularity and is the basic unit of a system which can be further sub-divided into elements [26, 27]. For example, robot is a component that is composed of elements such as motors, drives, *etc*. The '*station*' layer comprises of processing units that can assemble the workpiece. The '*pilot line*' layer represents a prototype line that is used for process and product validation at low volume. 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*' layer, however, does not cover the logistics and warehouse areas. At the highest level of abstraction is the '*factory*' layer that encompasses vehicle management systems, packaging units, warehouses and production lines. The modelling of factories and supply chains is beyond the scope of this research; the study is limited to the modelling of component, station, pilot and production line layers. The scale-up phase comprises of different dimensions such as hardware and software modifications, work culture changes, data management, automation of data transfer, etc. However, the research revolves around the hardware and software modifications associated with assembly system design during the scale-up phase; it is also assumed that there is an existing pilot or production line during the start of the scale-up project that can provide essential data regarding the considered processes.

1.4. Research contribution

The main contributions of this paper are highlighted as follows:

- A novel approach to replace the existing experience based scale-up process in industries is proposed. This helps to shorten the development and changeover time, enabling industries to maintain a strategic advantage over their competitors and ensuring the reduction of the cost, time and effort spent on the scale-up projects.

- A component-based bottom-up methodology for scale-up design support is formulated over four blocks of software: VueOne, Protégé, Flexsim and MATLAB, that are respectively used for process simulation, knowledge representation, production system simulation and optimisation. Consequently, the accuracy of the production system simulation in DES can be increased with data from knowledge-based kinematic modelling.

1.5. Structure of the paper

The remainder of the paper is structured as follows. Section 2 reviews the related literature on scale-up. Section 3 presents the overall architecture of the proposed Data-Driven Scale-up Model (DDSM) model with a detailed description of workstation configuration selection,ontology model generation and production line configuration selection. Section 4 details the multi-objective optimisation considered. Section 5 presents the case study and section 6 discusses the results and the validity of the approach. Section 7 concludes the paper and outlines the future work.

2. Literature review

2.1. Scale-up definition and characteristics

Scalability is regarded as a subset of reconfigurability and is related to changeable manufacturing and flexibility [22] and identified as one of the characteristics of Reconfigurable Manufacturing Systems (RMS) [28]. Closely associated with scalability is the term '*scale-up*', originating from computer-science background [29]. Another term akin to scale-up is '*capacity planning*' which, like scale-up, is associated with modifying the configurations of a system, both physical and logical, to accommodate the changes in demand [21, 4]. However, as seen from **Figure 1**, the capacity planning phase considers the daily demand changes and endeavours to meet the demand, primarily by modifying the operational policies and is characterised and influenced by the frequent but slight demand changes. No major hardware or software changes are generally executed and it does not comprise of production line stoppages since the scale of demand change does not warrant such practices. On the other hand, the scale-up phase is a critical project that is undertaken to make major modifications to the facility, both hardware and software. A minor modification to the control strategy or the manufacturing policy is insufficient to achieve the required demand following scale-up phase. The target demand for the scale-up project is usually a significantly higher number than the existing demand.

2.2. Scale-up and ramp-up

Ramp-up phase is defined as [2]: *"the time between the first part produced following system reconfiguration until reaching the required throughput level"*. In the manufacturing system life-cycle, ramp-up phase commences on conclusion of the concept development stage where process conception and development is done [30] and is primarily associated with New Product Introduction (NPI) and product changes. Depending on the industry and the phase of the system life-cycle, 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 major system modifications and ends on achieving the desired targets, the term scale-up primarily considers product volume increase. It is to be noted that ramp-up phase might be prolonged with additional adjustments to meet the targets 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 and hence the review on ramp-up as part of the literature survey.

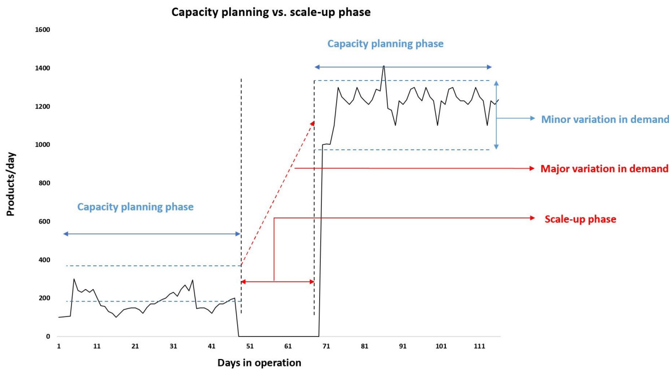

Figure 1: Capacity planning vs. scale-up

2.3. Existing knowledge on scale-up and decision-making

Several work in the pharmaceutical domain pursued by Levin [16], Faure [18], Tsinontides [17] and Wirges [31] discuss process industry scale-up. A decision support framework is proposed by Stauder [32] for technology selection in high volume production. Klocke et al. [6] proposed a framework wherein a hybrid simulation model using DES and system dynamics is used to support the ramp-up phase. Surbier et al. [33] summarised the characteristics of ramp-up phase and the problems faced during ramp-up. A simulation-based approach to plan for personnel during ramp-up is discussed as part of another research wherein a simulation-based algorithm using Plant Simulation and DES-based decision support are employed during the ramp-up phase [34]. In their paper, Almgren [35] identified the factors that affect the efficiency of ramp-up and in another research work, Ball et al. [36] identified a production ramp-up modelling framework. According to Colledani [2], an understanding of the disturbances that affect the system can lead to reduction of throughput losses during ramp-up by creating a system design that is robust. In another piece of work, three performance metrics which are the functionality, quality and optimisation are discussed to measure the progress of ramp-up [37] and a methodological approach for early identification and minimisation of scale-up risks is proposed by Elstner and Krause [38].

A plethora of papers discuss scalability of manufacturing systems with most of the papers focusing on the reconfigurable manufacturing systems. From a survey of related papers, the

Table 1: 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	Solution approach
1	Almgren [25]	2000	■					NA
2	Fricke & Schulz [23]	2005	■					NA
3	Deif & H.A. Elmaraghy [4]	2007					■	System dynamics
4	Deif & H.A. Elmaraghy [21]	2007					■	Genetic Algorithm optimisation
5	A.M. Ross et.al [24]	2008				■		NA
6	Guschinskaya et. al [39]	2008		■				Heuristic-based approach
7	Wang & Koren [40]	2012		■				Genetic Algorithm optimisation
8	Putnik et.al [22]	2013	■					NA
9	Bensmaine et. al [41]	2013			■			Optimisation using NSGA II
10	Ghani [42]	2013		■	■			DES & Kinematic modelling
11	Michalos et. al [43]	2015		■	■			DES-based hybrid approach
12	Renna [44]	2017					■	Gale-Shapely method
13	Manzini et.al [45]	2018		■	■			Simulation Optimisation (Linear Programming)
14	DDSM (presented approach)	2020		■	■			Multi-domain Simulation Optimisation (GA)

two important principles of implementing scale-up can be identified as i) linking or adding identical elements/stations to increase the productivity ii) increasing the performance of an element/station by changing its functionality [22]. In this regard, a method of quantifying scalability within the wider context of changeability is proposed by Ross [24]. A notable work that discusses cost modelling for scalability is proposed by Deif and Elmaraghy [21]. It provides significant pointers for scale-up but the actual method of performing the design modifications for scale-up is not discussed. In another related paper [4], Deif and ElMaraghy have assessed alternate strategies for different demand scenarios for RMS with the help of a System Dynamics model. Both the above-mentioned papers focus on capacity scalability and not on the scale-up planning phase which is discussed in this research work.

With regards to scalability planning and management at the production line level, Almgren [25][35] emphasised the importance of identifying disturbances, modelling failure and breakdown of workstations during the pilot phase. Wang and Koren have presented a GA-based optimisation algorithm that can help decision making about adding or removing machines from production lines in the event of a new market demand [40]. However, certain elements such as the material flow, labour, operational cost, space occupancy and the use of simulation models to better represent the complex production systems are not considered in detail. Additionally, the use of multi-objective optimisation over single-objective optimisation could provide the decision maker with more choices and flexibility. Hafezalkotob et al. [46] have tackled production planning and decision making across multiple plants using a game theory approach. On a similar note, Renna [44] has proposed an approach using the Gale-Shapley Model through which they support decision-making in reconfigurable workstations.

In the domain of workstation level scale-up, Bensmaine et al. [41] tackled the machine selection problem for RMS with a multi-objective optimisation method using Non-dominated Sorting Genetic Algorithm (NSGA - II). However, the approach primarily focusses on RMS, Reconfigurable Machine Tools (RMT) and machining operations, with a subjective candidate selection procedure. Manzini et al. [45] proposed a top-down approach designed around the Core Manufacturing Simulation Data (CMSD) standard to support production system design and reconfiguration. However, the article does not include a detailed discussion of production equipment and Material Handling Units (MHU) selection which are both vital for the scale-up phase. Kampker et al. highlight the benefits of DES during the planning stages to enable fast decision making and reduce time-to-volume [47]. Although a methodology to use DES to support the early developmental phase is provided and the modelling of scalable production system is described, the actual strategy of implementing scale-up is not clearly discussed.

2.4. Summary of literature review

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 perceivable from **Table 1**, a small number of papers discuss the selection of suitable technologies or solutions and the impact of technology changes on product quality. The solution approaches for the reviewed papers are also presented in the table and it can be seen that optimisation using GA is a popular solution approach. However, the difficulty associated with the mathematical modelling of complex manufacturing systems has resulted in the adoption of simulation-based optimisation. Another interesting stream of research is the application of game theory for supporting decision making in manufacturing systems; this is an avenue that will be explored in future works.

From the above mentioned review, the benefits of using simulation to support the system design is well-established and proven to reduce time to market. However, enough emphasis is not given to the selection of suitable system designs for a successful scale-up project. Additionally, a robust systematic approach for smooth transition during scale-up phase is not sufficiently explored. This is the research gap that this article aims to fulfill.

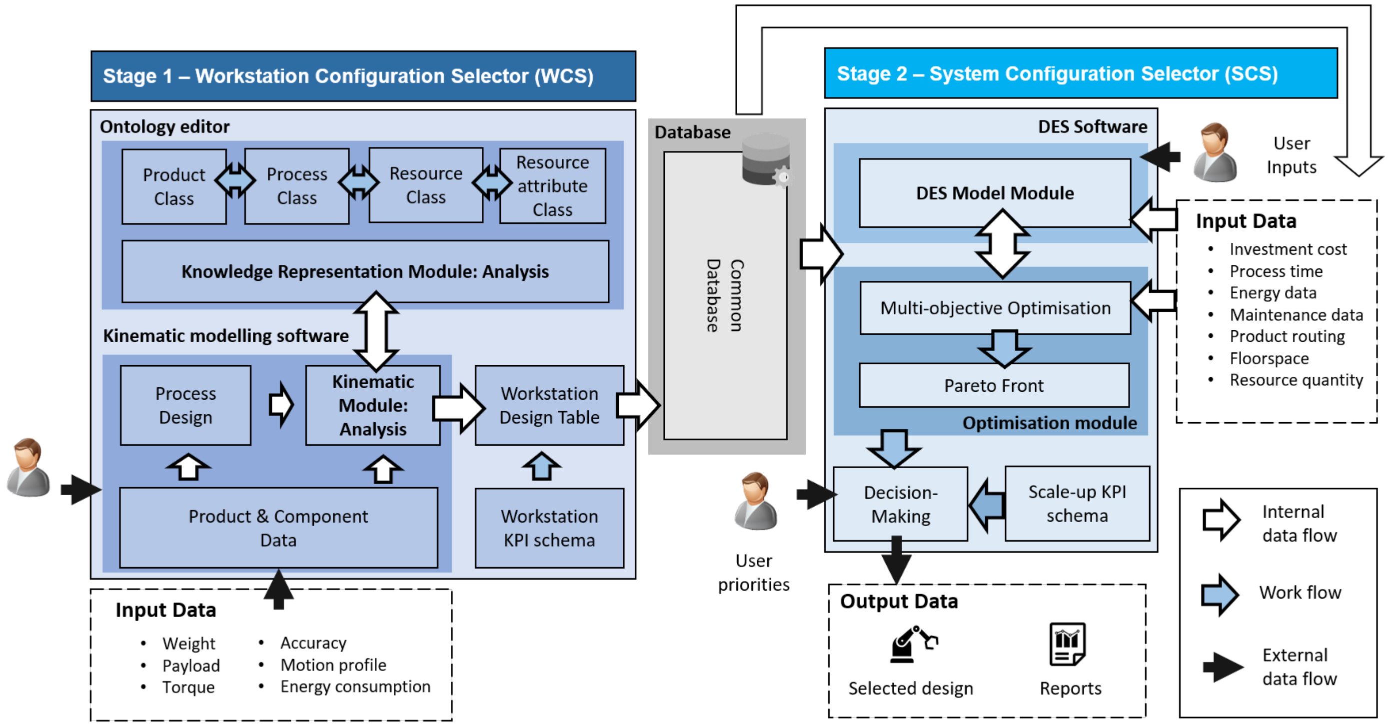

Figure 2: Architecture of DDSM

3. A data-driven scale-up model (DDSM)

3.1. Overview of the methodology

The methodology proposed in this article is termed as the Data-Driven Scale-up Model (DDSM) and is constructed upon two main pillars: *i*) Workstation Configuration Selector (WCS), also known as Stage one and *ii*) System Configuration Selector (SCS), also known as Stage two. The core idea behind the approach is to use digital manufacturing to identify system designs that could help realise scale-up. The concept is to leverage the data from workstation models in kinematic modelling software to improve the accuracy of system models in DES software to further perform meaningful analysis to support decision making during scale-up phase. However, the behaviour modelling of complex systems demands a broad spectrum of software, hence the importance of data integration. Although commercially available software platforms for digital manufacturing promise interoperability among a multitude of software, their capability to support heterogeneous software is not quantified. Regardless, this research study does not discuss issues related to software interoperability.

3.2. Architecture of DDSM

The architecture of the DDSM (**Figure 2**) is divided into two stages: *i*) generation of the workstation candidates and *ii*) generation of system candidates. Stage one is framed upon the assumption that the workstations that comprise of one or more equipment to perform the required assembly process can be categorised depending on the process that they perform. Thereby, there can be different types of workstations that perform similar, if not the same, process. Equipment for each workstation are selected within the knowledge representation module and the workstation KPIs are stored in the database. In Stage two, the workstation KPI data available in the database are accessed by the DES module for creating production system models. Thereby, data from the kinematic models at lower level of abstraction are accessible by the DES module models at a higher level of abstraction to subsequently improve the modelling accuracy. Successively, the DES model is coupled with the optimisation module, wherein, a multi-objective optimisation model for selecting near-optimal system designs for successful scale-up is employed.

3.3. Workstation Configuration Selector - Stage one

This phase is comprised of three elements: kinematic modelling module, knowledge representation module and the workstation design table (**Figure 3**). The kinematic modelling module is positioned within 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 and Resource Attribute (PPRR) framework adapted from the equipment ontology proposed by Ferrer et al [48]. The workstation design table comprises of the workstation designs, constituent equipment and workstation KPIs.

Table 2: KM component input

Data	Importance		Data format			Data source	
	Core	Optional	Numerical	Graphic	String	CAD	Datasheet/Vendor documents
Component CAD	■			■		■	
Mass		■	■				■
Payload		■	■				■
Torque		■	■				■
Direction of motion	■				■		■
Range of motion	■		■				■
Accuracy		■	■				■
Repeatability		■	■				■
Energy consumption		■	■				■
Cost		■	■				■
Acceleration/deceleration profile		■	■				■

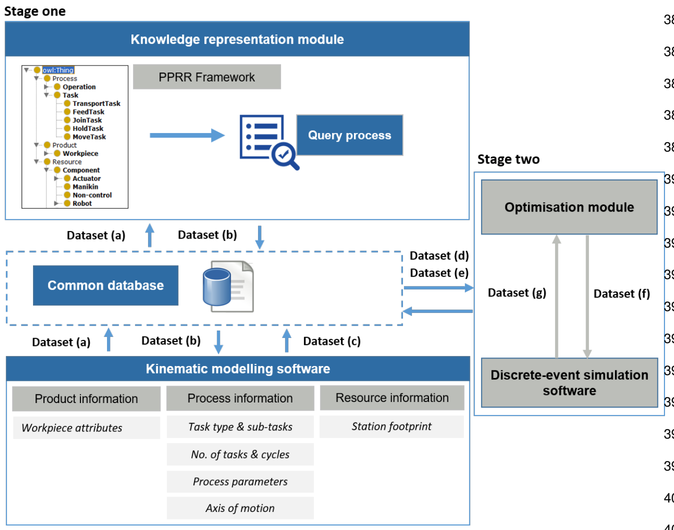

Figure 3: Workstation Configuration Selector.

3.3.1. *Kinematic modelling module*

Kinematic modelling software 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 [49]. Due to their ability to model the kinematics, they can predict the workstation processing time [50]. The workstation processing time data from the kinematic model can be leveraged to increase the accuracy of DES models [51]. Hence, in this research, a kinematic modelling tool, vueOne, developed by the Automation Systems Group in the University of Warwick, is used to model the existing pilot line to encapsulate within it, the product and process data available from the physical system. This model can then be utilised to perform future analysis on the behaviour and performance of potential workstation designs. This way, the concept designs of workstations can be virtually validated in the absence of a physical counterpart.

The data presented in **Tables 2** and **3** serve as typical inputs for process and workstation modelling. The architecture of the component and workstation databases are shown in **Figure 4**. It is important to note that the terms *'equipment'* and *'component'* essentially refer to the same object in this article. The component database consists of data regarding both non-control and control components, and the various parameters associated with them. Non-control components, such as the workstation frame, are not related to any process or action tasks and are generally stationary. However, it is important to model them since they support the visualisation and analysis of necessary workstation features such as station footprint, weight capacity, etc. which are useful to compare the workstation configurations. Control components, such as grippers and robots, typically perform tasks or actions and are associated with kinematics and process sequence. The *'lifecycle data'* from the physical system (the existing assembly line) and *'user inputs'* that deal with product and process data are embedded in the created kinematic model. Following the analysis of data, the process parameters, constraints, process sequence, machine setup, etc. are passed from the kinematic modelling module to the knowledge representation module. It is important to note that the above-mentioned process and product-related parameters might take up different values depending on the product variant that will be assembled. 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.

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 [52]. It was selected due to its wide and active user community, accessibility, availability of support and its potential to communicate with other research software [53, 52, 54]. Ontology, as explained by Gruber, *"is an explicit specification of a conceptualization"* [55].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 [56] and iv) to achieve data mapping be-

Table 3: KM workstation input

Data	Importance		Data format			Data source				
	Core	Optional	Numerical	Graphical	String	CAD	Process planning	Component library	Historical/ empirical data	Datasheet/Vendor documents.
Footprint	■		■			■				
Process seq.	■				■		■		■	
Safety interlock	■				■				■	
M/C setup		■	■						■	
Energy consumption		■	■					■	■	■
Work instruction	■				■				■	
Cost		■	■					■		■
Comp. motion time	■		■					■		
Position of states	■				■					
Layout	■			■		■				

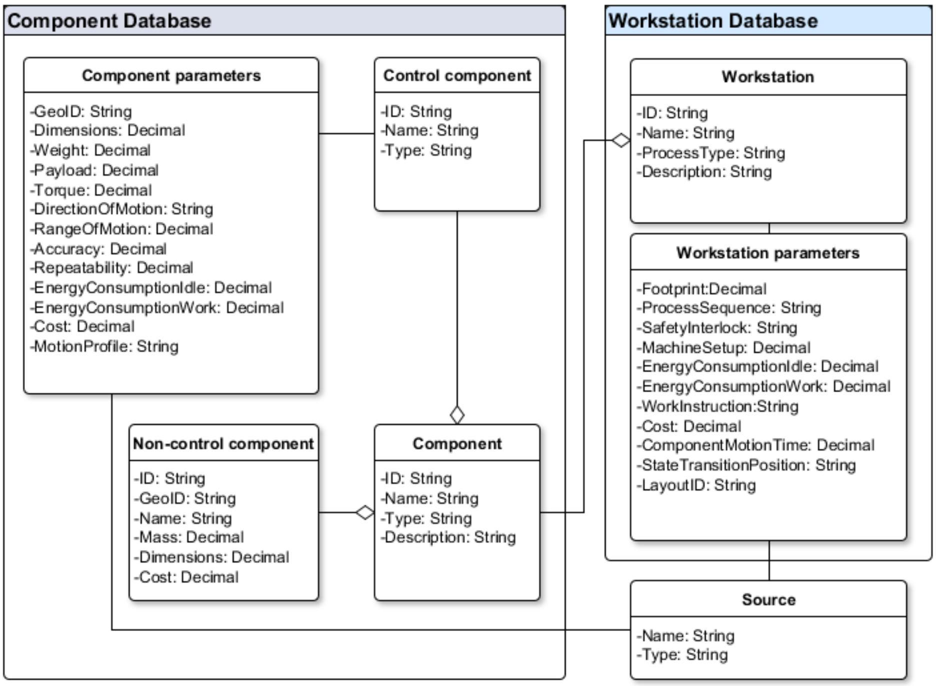

Figure 4: Component and workstation modelling architecture

tween heterogeneous software [57]. 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, endeavours to generate potential workstation configurations by retrieving, from an existing catalogue of equipment, suitable candidates 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 [58, 59].

The authors would like to highlight two previously published work on manufacturing ontology that are relevant to the research study. The first one is a PPR ontology adapted from [48], [60] wherein product attributes are mapped to process and resource concepts and integrated with a kinematic modelling software. The second work presents a Function Behaviour Structure framework for equipment selection [58]. The research proposed in this article is based on the PPR framework from [48] since both research works aspire the integration of the ontology model with a kinematic modelling software. The existing PPR ontology is improved with the novel addition of i) the 'resource attribute' class ii) data properties that relate to process parameters for assembly operations and iii) query design for workstation configuration selection. Additionally, the proposed methodology differs from the mentioned articles in that it pur-

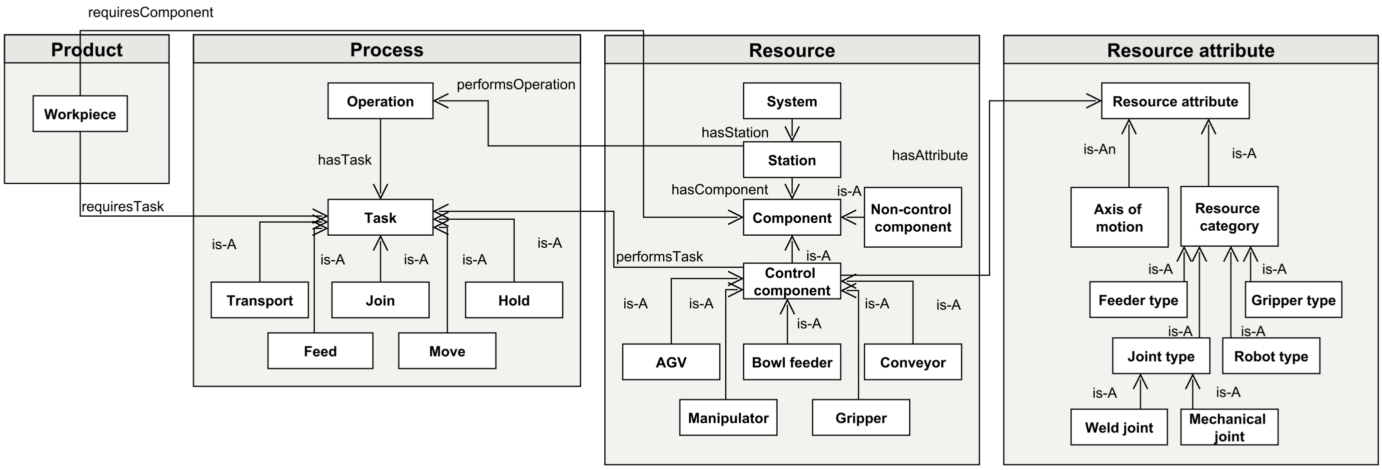

Figure 5: DDSM ontology structure in protégé.

sues the objective of supporting system configuration selection for transition from low-volume to high-volume. Another point to note is that the DDSM approach is not necessarily bound to system reconfiguration but also considers the commissioning of new facilities and replacement of existing workstations that are unfit for purpose.

The presented ontology framework comprises of product, process, resource and resource attribute classes as explained in **Figure 5**. The product class comprises of a *'workpiece'* or *'part'* that is mapped to a resource as well as the required assembly process. The process class comprises 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 operation sequence is represented as a state transition diagram (STD). For the purpose of this research, five task types, move, hold/release, feed, transport and join, adopted from [27] are considered. In the ontology, the five considered tasks are represented as five instances that belong to the *'task'* subclass.

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 in turn built up of components. The term component here refers to the equipment such as weld gun, robots, etc. that are used to perform various tasks. Components are further subdivided into control and non-control component, as shown in **Figure 4**, depending on whether they have logical behaviour or not. Five types of control components are considered in this study: 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'* having certain parameters can be added as an *'instance'* to the gripper subclass. In this way, the various components are added to their corresponding sub-classes as *'instances'* and mapped to one or more of the defined five tasks using the *'performsTask'* object property; object properties are used to relate or map two instances or individuals. 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. The *'move'* and *'feed'* are instances of the *'task'* subclass and robot *'ABC'* is mapped to the two tasks using the *'performsTask'* object property.

On the other hand, data properties are used to map an instance to a specific type of data that can be real number, integer or string. The values of data properties such as range, dimensions and payload of the resource elements can be obtained from the component and workstation database, the architecture of which is illustrated in **Figure 4**. 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.

The resource attribute class consists of two sub-classes: *i*) axis of motion, which includes the assembly directions of the resource and *ii*) 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 axes of motion and category.

The kinematic model is analysed to obtain the operation sequence and the total number of operations. The operations are identified as *'O'* and the total number of operations is '*No*'. The tasks are identified from the operation sequence. According to the flow chart in **Figure 6**, the tasks that belong to each operation are verified to understand whether they belong to one of the defined task types. If they do, then the operation is explored in detail within protégé to identify suitable workstation designs. Otherwise, the operation will be ignored and the next operation in sequence will be subjected to the same procedure. Within protégé, the requirements for each operation are first considered and translated to parameters that are used to screen the existing set of components using *'query'* to identify suitable equipment. The process of information retrieval is done with the help of the query language, *SQWRL* (Semantic Query-enhanced Web Rule Language). This enables the screening of a catalogue of equipment, those defined as instances within component subclass

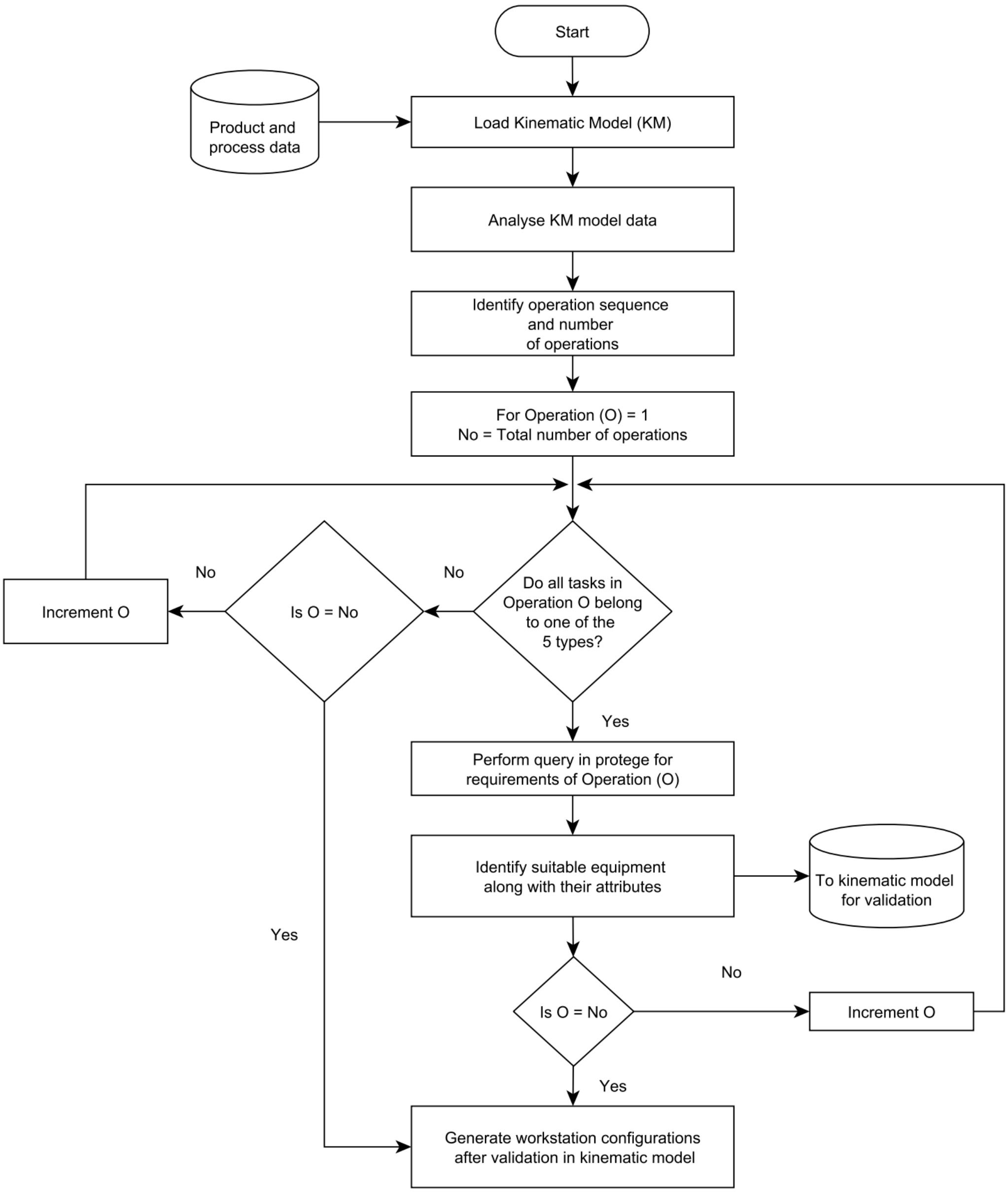

Figure 6: Knowledge representation module process flow.

of the resource class, to find the components that are suitable to perform the required tasks. A demonstration of the query process is provided in section 5.2. 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 further modelled in the kinematic module. For instance, after performing query, the selected equipment might meet all the required parameters, but in reality 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 process time can also be calculated. In this regard, a previous work done by the author highlights the benefits of the integration of kinematic model and ontology module [51].

To summarise the characteristics of the knowledge representation module, it is necessary to explain how the kinematic model and ontology editor 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 module. It is difficult to calculate the workstation process time, investigate collision detection, perform path planning and ergonomical analysis, and check assembly feasibility within the knowledge representation module. These issues can, however, be overcome by using the kinematic model module and hence the coupling of both modules improves the accuracy of workstation modelling. An important

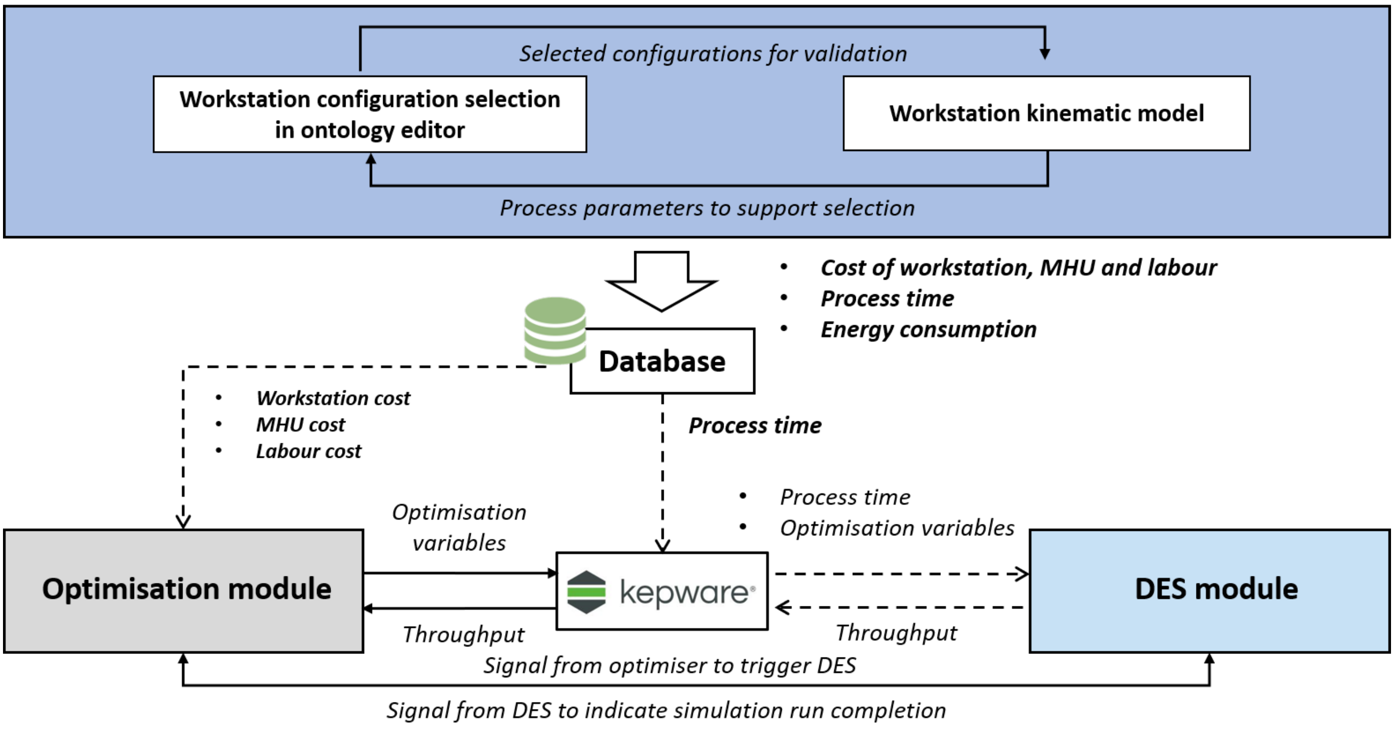

Figure 7: The integration of the the optimisation module and DES module using OPC-UA based communication

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 table*

The workstation design table comprises of the workstation KPI schema which serves as a template to create the workstation design matrix. The design table consists of the workstation candidate configurations that were validated in the kinematic model. The workstation configuration is defined by the identified equipment that are suitable for the considered processes and the corresponding metrics such as investment cost, process time, energy consumption, geometry and Computer-Aided Design (CAD) information of the workstation. This data is stored in the workstation database such that it is accessible by the software used in Stage two of the methodology.

3.4. *System Configuration Selector - Stage two*

The primary aim of Stage two, shown in **Figure 7** is to identify potential assembly line configurations with the help of two modules, the *'DES Model'* module and *'Optimisation'* module, that facilitate simulation optimisation for stochastic discrete-event systems. DES is increasingly used in the field of manufacturing for building models that allow the comparison of alternate scenarios, answering *'what-if'* questions and supporting decision making [61, 62, 63]. The inherent capability of DES to model production systems is a key reason to use it as part of the methodology. 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 (Stage one) to increase the accuracy of DES [51].

Optimisation is the process of finding one or more solutions that either maximise or minimise the formulated objective function whilst satisfying the defined constraints [64]. 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 [65].

3.4.1. *DES model module*

The workstation KPIs from Stage one are conveyed to the DES model to increase the accuracy and transparency at the system modelling level. Additionally, certain user inputs such as the consideration of model abstraction, simulation graphic settings and parameters, representation of the process logic, etc. are also required as seen from **Figure 2**. In a previously published work by the author [66], pilot line scale-up using DES to investigate the impact of scheduling policy and scale-up principles on certain system level KPIs was demonstrated. As part of the paper, additional stations and configuration changes were

Table 4: 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	capital 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	capital 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'

implemented in the virtual model. However, it was not possible to analyse the practicality of such solutions using DES alone. This was due to top-down approach using a standalone DES model and hence the lack of access to the workstation-level data. This is a vital issue that will be addressed in the following sections.

3.4.2. Optimisation module

Genetic Algorithm (GA) is a popular meta-heuristic numerical evaluation method that is compatible with DES for simulation optimisation [67]. The optimisation module is coded in MATLAB and the *'gamultiobj'* solver which uses a controlled elitist GA is employed. The controlled elitist GA favours diversity which is considered essential for convergence. The 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 set of trade-off solutions are expressed on the pareto front; the pareto front plays a key role in decision making along with inputs from user. Additionally, the scale-up KPI schema provides criteria for decision making. The selected system configurations encompass the workstation configurations along with the various reference IDs that are used to address the process, workstation, components, etc. Consequently, they are stored in the system database and intended to support the scale-up planning in industries.

3.4.3. Data exchange between DES and optimisation module

In order to support the ease of data transition between the DES and optimisation modules, the DES model needs to be parametric such that it can reach the desired level of configurability and adaptability which is considered vital for simulation optimisation [68]. The parameters need to be modified from outside the DES model automatically. In order to achieve this integration, it is important to consider three main elements i) the interface for data transfer ii) the method of sending and receiving data from MATLAB iii) the method of sending and receiving data from DES.

For the interface, KEPServerEX software which provides real-time data transfer with the OPC UA communication protocol is used. 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 tags in the server hold the values of the decision variables which are passed to FlexSim and throughput values that are passed back to MATLAB. 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 communication between MATLAB and the server is established using *'OPC toolbox'* in MATLAB. At the end of the simulation run in FlexSim, the throughput values are passed back to MATLAB using the server. Depending on the software used for the DES and optimisation, this procedure might vary. However, the underlying approach and objectives remain similar.

4. Problem formulation

4.1. Objective functions

The considered mathematical notations are given in **Table 4** and it is to be noted that workstations are categorised according to the operations performed and all workstations that perform the same operations belong to the same type (represented as 'w'). For some of the considered workstation types, different alternatives performing the same operation are identified in

Stage one and referred to as workstation configurations. The alternate configurations for a particular workstation type constitute a decision variable in the optimisation module.

The considered optimisation problem has two conflicting objectives, *i*) scale-up cost which is detailed in **Equation 1** and *ii*) system throughput which is detailed in **Equation 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 MHUs of each type required and iv) identify the suitable configuration for workstations such that the required throughput can be achieved while within the scale-up budget.

Objective 1 is the scale-up cost which consists of four main elements. Due to confidentiality reasons, scale-up cost is represented in *units*. 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 which is represented as slots within which workstations can be added. If the space restriction is not violated, then the penalty cost, β, is zero. However, on violation of the space constraint, the penalty cost is calculated to be a value greater than zero.

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

Please note that the direct and indirect raw material costs, indirect labour costs and maintenance costs are not considered in this objective function as they are assumed to be a constant across the iterations.

Objective 2 is to maximise the system throughput.

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

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.

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.

4.2. *Assumptions*

- The station footprint of all workstations are assumed to be equal in size.
- 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 period of simulation.
- The simulation model does not include warehouse and other industrial departments; only production line and associated operations are considered.

4.3. *Proposed GA method*

The proposed optimisation method utilises the multi-objective mixed-integer GA, which is a heuristics-based evolutionary algorithm. In GA, the individuals of each generation comprise 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. 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* [65].

The pareto front population fraction which determines the number of solution points on the pareto front is 0.35 by default. 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, '*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, *ii*) stall generation limit, and *iii*) maximum time limit. The pseudo code for the GA is given in **Table 5**.

It is important to ensure that the individuals represented on the pareto front should be diverse enough to represent the range of pareto front. The pareto front solutions around the *'knee'* of the front, exhibit acceptable fitness scores for both the objectives considered. The selection of candidates from the pareto front requires an evaluation process using the decision maker's priorities and inputs which is demonstrated in the case study. The workflow of the data transfer is provided in the following steps and the details regarding the MATLAB and FlexSim codes and functions are provided in the next section.

Table 5: 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;

1. The initial set of values for decision variables are decided in MATLAB 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 for every optimisation iteration using a *'batch file'*.
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 two.
6. The workstation cost, material handling unit cost and operator cost are accessed from the database by MATLAB to calculate objective function one.
7. 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.

5. Implementation

5.1. Description of the assembly line

The methodology is implemented in a pilot production line that assembles two variants of battery modules, A and B. Most operations are common across both variants with some variations present in the welding and cooling system assembly. The considered case study has eight operations, as shown in **Table 6**; operations one and two are explained in more detail to demonstrate the use of knowledge representation module. Variant A comprises of 120 cylindrical *'18650'* cells while variant B has 90 cylindrical *'21700'* cells. The testing (operation one) and cell loading operations (operation two) are performed in workstation one. The thermistor (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. Considering the allocation of the operations to workstations, there are five workstation types in total; workstation type four is bypassed by variant B since it does not require pulse-arc welding. Similarly, workstation type five is bypassed by variant A since it does not require wire bonding. Variant B also has a different cooling system assembly due to the inherent difference in the module design in comparison to variant A. The transfer of products between stations is achieved with conveyors; buffers to store products between stations are not available. The product designs are confidential and will not be explained in detail. An image of the pilot line facility at the University of Warwick is presented in **Figure 8**. The case study starts with the modelling and encapsulation of data pertaining to the five workstation types in VueOne. The target daily demand that is considered is 65 products of A and B while the current daily production volume is 20 products of A and B.

Table 6: Allocation of operations to workstations

Station number	Operation number	Operations 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

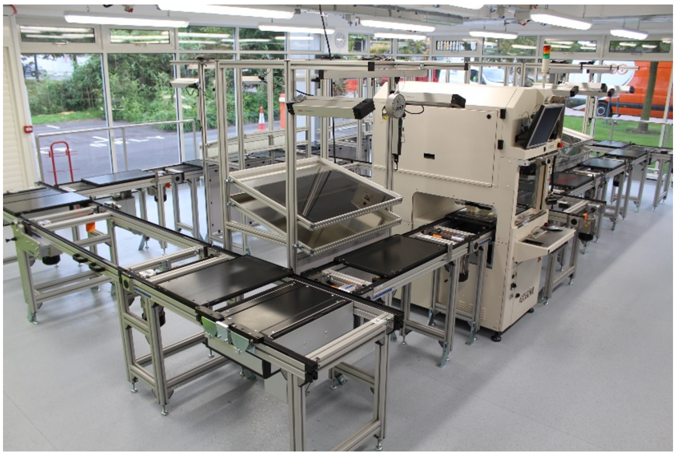

Figure 8: Pilot line for electric vehicle battery module assembly.

5.2. Demonstration of methodology: Stage one

The first two operations from **Table 6** are explored in detail to demonstrate the implementation of the workstation configuration selection process. The task sequences for operations one and two are provided in **Tables 7 and 8**, respectively. The task sequence for both product variants is the same with slight variations in the positioning due to the geometrical differences in the cells. The first step is to model the operations in vueOne. For operation one, which is the testing operation, the vueOne model replicates the tasks performed in the pilot line using the CAD of the cell testing equipment 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'*. For operation two, which is the cell loading operation, the V-Rob module is used; the V-Rob module has a pre-defined library of robots from which a general purpose ABB robot is selected. The gripper CAD is imported into the model and the translation kinematics 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 side and loads them into the battery module. The kinematic model of operation two is presented in **Figure 9**.

Table 7: Task sequence for operation one

Task number	Task name
1.1	Move to position (x1,y1,z1)
1.2	Test battery cell (*30)
1.3	Repeat steps 1.1 & 1.2 with offset of 120mm in x

Table 8: Task sequence for operation two

Task number	Task name
2.1	Move to position 1
2.2	Hold battery cell
2.3	Move to position 2
2.4	Release battery cell
2.5	Repeat steps 2.1 to 2.4 with offset 20mm in x
2.6	Move to position 3
2.7	Hold battery cell
2.8	Move to position 4
2.9	Release battery cell
2.10	Repeat steps 2.6 to 2.9 with offset 20mm in x
2.11	Move to position 5
2.12	Hold battery cell
2.13	Move to position 6
2.14	Release battery cell
2.15	Repeat steps 2.11 to 2.14 with offset 20mm in x
2.16	Move to position 7
2.17	Hold battery cell
2.18	Move to position 8
2.19	Release battery cell
2.20	Repeat steps 2.16 to 2.19 with offset 20mm in x
2.21	Move to position 9
2.22	Hold battery cell
2.23	Move to position 10
2.24	Release battery cell
2.25	Repeat steps 2.21 to 2.24 with offset 20 mm in x

The next step is to use the flow chart in **Figure 6** to consider and assess the eight operations that are distributed across the workstations. For operation one, there are two tasks, 1.1 which is a move tasks and 1.2 which is a test task. The test task does not fall within the defined five tasks and hence operation one will not be considered for further analysis. Advancing to operation two, the cell loading operation has two types of tasks which are the move, hold/release tasks. Both tasks are within the defined list and hence operation two is considered eligible for further analysis. Progressing to the next step in the flowchart, 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 kinematic model for performing the query.

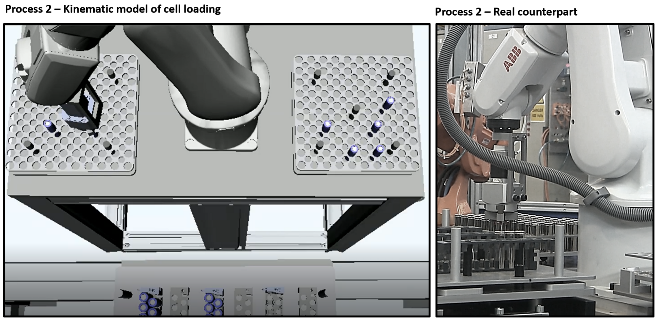

Figure 9: Kinematic model of the cell loading operation.

One query is designed for each operation, considering the parameters for both product variants. Operation two contains the move and hold/release tasks; the query is designed in three sections as seen in **Figure 10**. 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. The '*sqwrl:makeSet*' function is used to create two sets, one for each task and the '*sqwrl:union*' is used to combine both sets together.

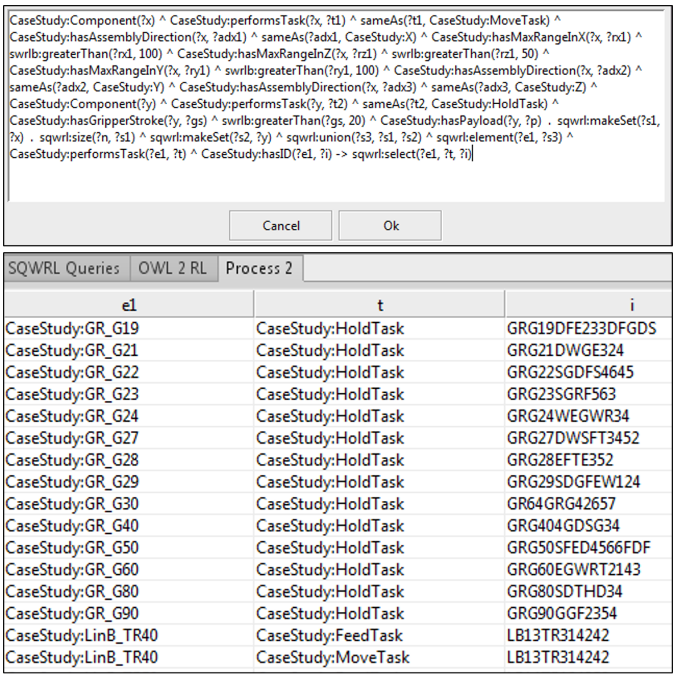

Figure 10: Query design and results

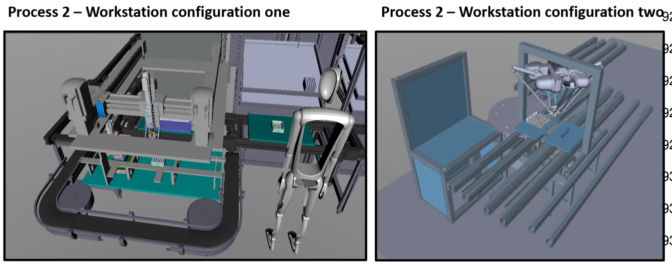

Figure 11: Modelling the workstation configurations in vueOne kinematic modelling software.

For the move task, the degree of freedom and working range are considered as primary criteria for equipment selection; for the hold/release task, the payload and gripper stroke are considered as primary criteria for gripper selection. The components from both sets, move and hold/release, are combined to provide a total of 30 components that meet the defined criteria. Two configurations are selected from this component list and validated as shown in **Figure 11**. For workstation configuration one with an ID of *'WS1GRG24LB13'*, a vacuum gripper and gantry with respective IDs of *'GRG24WEGWR34'* and *'LB13TR314242'* are selected. For workstation configuration two, with an ID of *'WS1GRG28LB15'*, a vacuum gripper and delta robot with respective IDs of *'GR4668HTDSD3'* and *'DB434DGSH'* are selected. Both workstation configurations are associated with their respective workstation KPIs using the references IDs.

The new configurations are visualised and checked for potential issues such as collision by modelling in vueOne. Following this, process time calculation is done with data available from datasheet, physics-based model, machine learning algorithm or experience based results [51]. It is also possible to calculate the time values using the capabilities of vueOne. It was found that the time taken to perform 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. In order to get the total workstation processing time of workstation one, the cell testing time needs to be considered since both cell testing and loading are performed on the same workstation. Therefore, the current cell testing time of 30 seconds is added to the time taken for operation two to get the total process time. It should be noted that it might be necessary to combine the process times of two or more processes to obtain the total workstation process time. The calculated time values and workstation configurations are ultimately intended for use in Stage two and hence stored in the workstation database along with their respective ID information, geometry, cost and maintenance data as seen in **Figure 12**.

5.3. *Demonstration of methodology: Stage two*

In Stage two, the system configurations need to be generated while also considering the workstation configurations chosen in Stage one. Firstly, a parametric DES model is created with workstations, AGVs and operators as shown in **Figure 13**. The distance between workstations is assumed to be a constant and the production line layout for the optimisation problem is assumed to be rectangular. Although the initial pilot line utilises conveyors for transportation, for the scale-up scenario modelling, AGVs are employed due to their flexibility to cater to more than one workstation. In the DES model, the eight operations seen in **Table 6** are allocated to five workstations. The assembly system is assumed to be mixed model 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 stations; stations can be bypassed if product variant does not need to be processed in a particular station. Buffers are allocated between workstations and can hold a maximum of five products; first-in-first-out (FIFO) scheduling policy is considered for the transfer of products from buffer to workstations. Each of the five workstations assemble only one product at a time. Similarly, AGVs can transport only one part at a time. Each workstation has a 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 product assembly is 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, the AGV and operator are free to work on any workstation and are not restricted to a particular region of the production system. Operators are assumed to be multi-skilled and both operators and AGVs are assumed to be always with the exception of break times. Stochasticity is introduced in the model using the 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, but part arrival follows the exponential distribution. These distributions are obtained based on literature and data from pilot line. A warm-up time of 2500 seconds is found suitable for the model; the total simulation time that represents a single shift is 28800 seconds and only one shift is modelled. The subcomponents and raw materials required for the assembly are assumed to be always available.

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 workstations are allocated to slots and workstations of the same type are 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 and this is represented using five decision variables, one for each workstation type. Each variable value refers to the quantity of the respective workstation type. For instance, if the second decision variable has a value of two, it means that workstation type two has another copy in parallel that performs the thermis-

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 batte1		X	Y	Z	Move,Test	100	100
Operation2	P2458HJ	Loading battery2		X	Y	Z	Move,Hold,Release	100	100

Resource data in database

ResourceName	ResourceID	Description	Footprint	DimensionUnit	AllowableWei	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 12: Workstation design table and workstation configuration data in the common database.

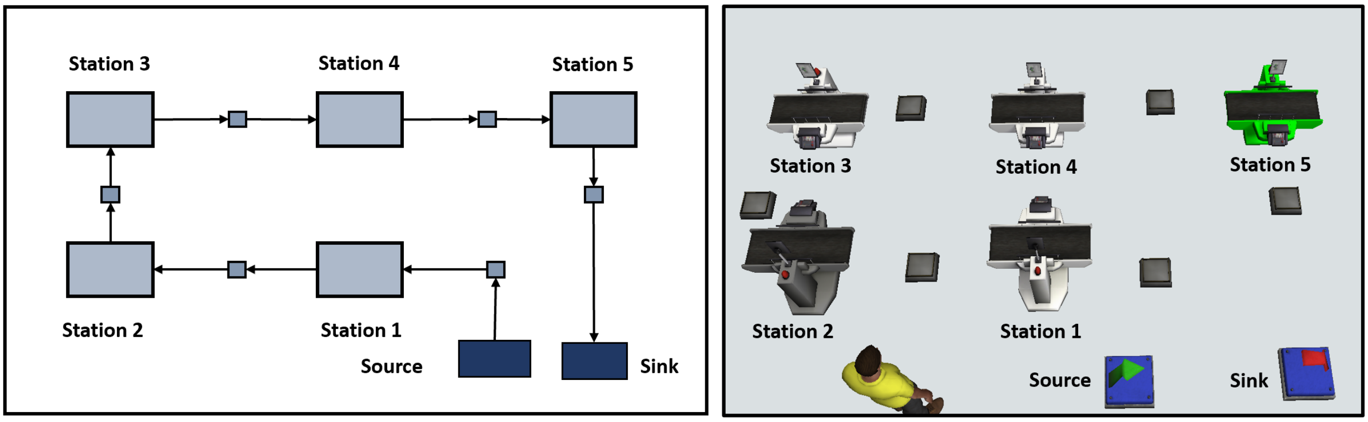

Figure 13: Flexsim model of the pilot battery module assembly line.

tor assembly (process 3) and cooling system assembly (process 4).

5.3.1. *KEPServerEX to FlexSim connection*

As explained in section 3, KEPServerEX is used to pass variable values to FlexSim. 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.

5.3.2. *MATLAB to KEPServerEX connection*

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. The algorithm settings include a population size of 20 and a maximum generation limit of 100, with ten simulation repetitions for each evaluation and a stall generation limit of 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.

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. 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 typically minimises the objective functions, objective two is rewritten as (1/throughput).

1. A vector called *'flexin'* that has the values of the eight decision variables is the input for this function. **Equation 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 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 will be described later, 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 **Equation 2**.
6. A new group and two item objects are created using MATLAB OPC toolbox 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.

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 which performs the testing and cell loading process and hence only workstation one has alternate configurations; therefore, the value of the variable l is one.

Table 9: 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	Configuration for workstation type 1	1	2

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 9**. 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.

Now that the core optimisation algorithm is defined, the *'FlexSim initialisation'* function function is written in MATLAB to support the core function. It is used to initialise the *'batch file'* that starts the simulation. The pseudocode for the FlexSim initialisation function is shown in **Table 10**. 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 10**, 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

Table 10: 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);**

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. **Figure 14** provides some shows the communication elements such as the tags, emulator elements and batch file.

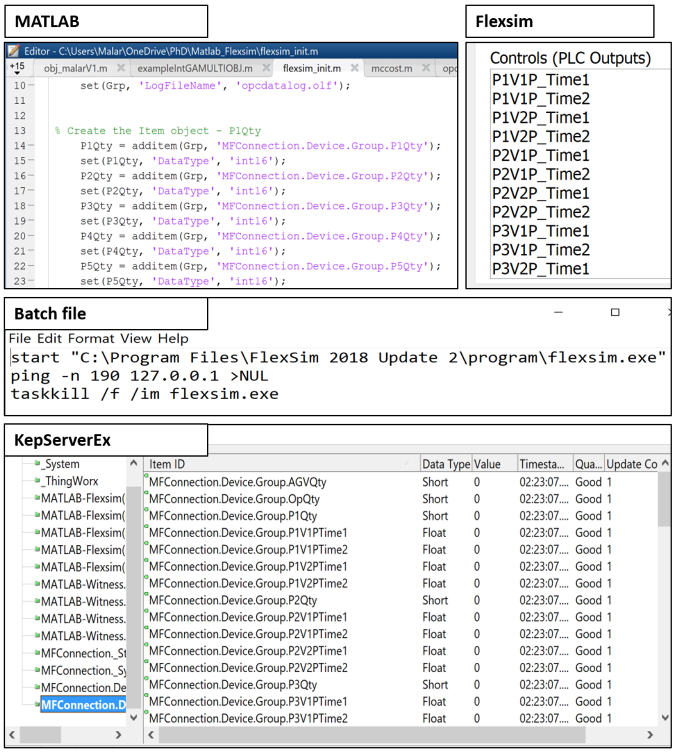

Figure 14: MATLAB and FlexSim integration.

6. Results

The simulation optimisation is achieved using a laptop with Intel Core i7 with a processor speed of 2.60GHz. 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. **Figure 15** shows the trade-off solutions obtained as a result of the multi-objective optimisation. 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. From analysing the results, it is identified that the data points indicated with the *‘asterix’*, which are located towards the left end of the graph, do not meet the required throughput and hence will not be considered for further analysis. Four

Table 11: Trade-off solutions selected for further analysis.

Solution	x_1^1	x_2^1	x_3^1	x_4^1	x_5^1	x_1^2	x_1^3	x_1^4	Obj 1 [Cost(units)]	Obj 2 $[1/(\varepsilon_1+\varepsilon_2)]$	Throughput A (ε_1)	Throughput B (ε_2)
1	1	2	2	1	2	1	4	1	65520	0.0064	77	78
2	1	2	3	2	2	1	5	1	74000	0.0051	100	93
3	2	2	3	2	3	2	5	1	98400	0.0046	105	112
4	2	3	4	2	3	2	5	1	112100	0.0045	108	114

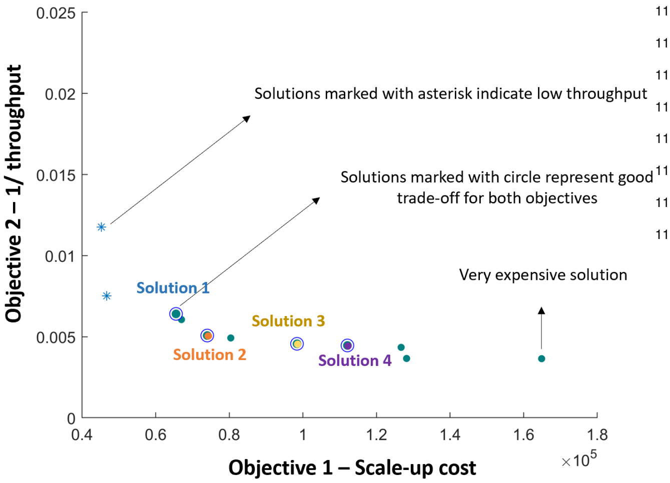

Figure 15: Non-dominated solutions for the battery module assembly case study.

solutions indicated with the *'coloured circle'* are selected for further analysis as they provide reasonable trade-off between scale-up cost and throughput. The selected solutions are tabulated in **Table 11**. They are further subjected to *'a posteriori'* approach, where the preferences from the decision maker are used to select a suitable solution from the considered list. 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. In this context, the ease of transition represents the time and effort taken to change the system configuration from the existing one to the proposed one.

6.1. Decision making from industrial perspective

A radar plot is provided in **Figure 16** 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 the target throughput of 65 products of variant A and 65 products of variant B. However, solution four has more production capacity than required. Depending on the application and scenario under consideration, the decision maker might consider solution 4, i) if the production line is intended to be used over a long period of time and ii) if the demand is expected to increase again in the future. Despite the 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 above situations. However, the solutions one and two might be considered i) for production lines that have relatively shorter lifespan or ii) for products predicted to become obsolete in the near future.

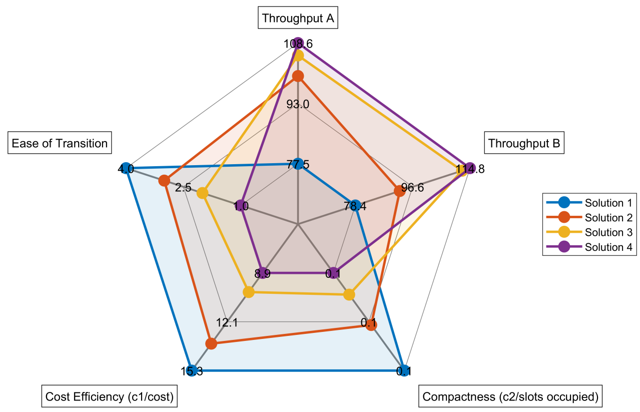

Figure 16: Comparison of solutions using radar plot

For the purpose of comparing the alternate DES scenarios, the total simulation time is considered as a constant and it is assumed that infinite number of products are available for processing. As a result, it is difficult to compare the solutions with regards to scheduling policies that prioritise one product over the other. For such cases, 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.

7. Discussion and future work

The calculation of workstation process time in Stage one of the methodology is achieved with the help of a kinematic model. It is also possible, with proper ontology and query design, to achieve the same using protégé for certain workstations, as long as the motion time information of actuators and such can be sourced from either datasheets, historical results, experience, etc. This approach can be considered more beneficial since it is capable of filtering only those workstation designs that fall within the preferred bounds of process time. Hence, the number of candidate solutions that need to be modelled virtually can be significantly lowered. Although similar approaches for selection of equipment from an available list of products for

Table 12: 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

specific brands are available in some manufacturer's website, a screening process across a wider range of brands might be more beneficial and time-saving. The reason behind this is that manufacturing industries, typically, employ a myriad of brands across their production systems.

The workstation configurations that are considered in the case study are mostly automatic or semi-automatic. In situations where manual workstations are involved, it is not necessary to perform the equipment selection process in protégé. The simulation optimisation process in Stage two has certain disadvantages due to the computation time as a result of the real-time connection, number of decision variables considered and replications within DES. Additionally, the process of triggering FlexSim from MATLAB negatively influences the computation time. Although the use of a batch file with a *'timeout'* allows the termination of Flexsim at the end of simulation run, the use of KEPServerEX to transfer data in real-time and issues with the quality of the connection, delay the execution process. The extra lines of code written in the algorithm for objective function calculation to ensure that good quality data is passed from FlexSim to MATLAB also prolongs the simulation optimisation time. To overcome this, *'a priori'* approach, where the decision maker's preferences are considered before performing the optimisation in order to focus the solutions in a particular region of the pareto space can be considered. An evaluation of the industrial applicability of DDSM is provided in **Table 12** where seven different criteria are considered. Accordingly, the methodology is found beneficial in providing cost, time and effort-savings along with supporting the reusability, extendability and traceability. The process of DES modelling demands some background knowledge about the considered application and production system as there is need to abstract and model the system in an efficient way. Also, several parts of the methodology in its current state, require human intervention. However, it is possible to achieve seamless integration in a more effective way using i) plugin for data transfer between the ontology editor and database for equipment selection ii) 'xml' files for automatic transfer of process parameter from kinematic model to the ontology editor iii) functions within MATLAB to obtain information from database and write back to it. Associated papers that discuss the connectivity between ontology model and database can be found in literature [69, 70, 71]. Currently, the selected solutions comprise of information about the quantity of operators, machines and material handling units but not about the layout. Further work needs to be done in the area for layout analysis of the selected solutions and improving the DES model to automatically place the newly added elements in the new layout.

In the case study implementation, four solutions from the pareto front were selected for further analysis using five scale-up KPIs which focus on the system design. However, it is possible to consider other criteria from operational perspective such as the machine utilisation, buffer usage, machine blockages, etc. The four solutions were further analysed in the DES model to check for potential bottlenecks and deadlock situations. Although not discussed as part of this research, there are plans to provide an improved version of the decision making process using multi-criteria decision making techniques along with the consideration of additional criteria about the selected solutions in terms of machine maintenance, breakdown and energy consumption. The proposed methodology could also be extended for warehouse modelling and complex material handling scenarios.

Table 13: Comparison of DDSM methodology with similar works.

	Related works			
	Ghani et al. [42]	Michalos et al.[43]	Manzini et al. [45]	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

7.1. Comparison with related works

The DDSM approach is closely related to the work done by Ghani [42]. The research work done on the integration between Kinematic modelling software and DES proposed by Ghani [42] 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. [43] and Manzini et al. [45]. Both these works focus on the system configuration and design problem. While Michalos et al. [43] support the robotic workstations using a two-stage approach combining analytical method and virtual modelling, Manzini et al. [45] 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 13**.

7.2. Conclusion

This research study proposes a methodology to support decision making for transition from low to high volume manufacture in a systematic way. This is achieved by data integration of virtual engineering tools that specialise in production line and process modelling. To support the transition phase, it is essential to understand the number of operators, new workstations and material handling units that are necessary for the new system design in addition to ensuring that the required throughput is achieved while still meeting the budget constraints. Therefore, two conflicting objectives, cost and throughput are considered for an evolutionary multi-objective simulation optimisation using MATLAB and FlexSim. The input parameters for the DES model such as the number of machines, operators, material handling units are passed from MATLAB to FlexSim through KEPServerEX and throughput of products A and B are passed back to MATLAB. Each workstation in DES is linked to a process simulation model to obtain the workstation processing time, subsequently improving the accuracy of DES and ensuring that the time values are feasible and realistic. The approach is demonstrated using a case study of battery module assembly and the multi-objective simulation optimisation along with the process simulation model provides potential system design solutions that are represented on the pareto front. The alternate solutions are compared according to five criteria that represent the scale-up KPIs and the pros and cons of each are discussed with the final decision left at the hands of the system designer.

Acknowledgment

The authors would like to thank Arun Gopal, Kaniappan Periakaruppa, Mus'ab Ahmad, Hala Abdelaziz, Rohin Titmarsh and Carlo Ferri for evaluating the industrial benefits and reviewing the DDSM methodology. The presented work was conducted as part of research studies through the project AMPLiFII 2 (InnUK/104175/1). The primary objective of the research is to provide a decision support framework to scale-up the battery module assembly facility present at WMG, University of Warwick.

References

[1] H. A. ElMaraghy, Flexible and reconfigurable manufacturing systems paradigms, International journal of flexible manufacturing systems 17 (2005) 261–276.

[2] M. Colledani, T. Tolio, A. Yemane, Production quality improvement during manufacturing systems ramp-up, CIRP Journal of Manufacturing Science and Technology 23 (2018) 197–206.

[3] C. Terwiesch, R. E. Bohn, Learning and process improvement during production ramp-up, International journal of production economics 70 (2001) 1–19.

[4] A. M. Deif, H. A. ElMaraghy, Assessing capacity scalability policies in rms using system dynamics, International journal of flexible manufacturing systems 19 (2007) 128–150.

[5] B. Alkan, An experimental investigation on the relationship between perceived assembly complexity and product design complexity, International Journal on Interactive Design and Manufacturing (IJIDeM) 13 (2019) 1145–1157.

[6] F. Klocke, J. Stauder, P. Mattfeld, J. Müller, Modeling of manufacturing technologies during ramp-up, Procedia CIRP 51 (2016) 122–127.
[7] B. Alkan, R. Harrison, A virtual engineering based approach to verify structural complexity of component-based automation systems in early design phase, Journal of Manufacturing Systems 53 (2019) 18–31.
[8] E. Szczerbicki, Simulation modelling for complex production systems, Cybernetics & Systems 31 (2000) 333–351.
[9] C. McLean, S. Leong, The role of simulation in strategic manufacturing, in: Proceedings of the 33nd Conference on Winter Simulation, Citeseer, pp. 1478–1486.
[10] H. S. Chemicals, Driving Lab Performance and Productivity, Technical Report, Germany, 2018.
[11] Kinaxis, An integrated approach to global capacity management for automotive manufacturers, Technical Report, Canada, 2018.
[12] M. Kinzoku, Electrodeposited Copper Foil Plant in Malaysia to Significantly Enhance Production Capacity, Technical Report, Malaysia, 2008.
[13] M. Kinzoku, Mitsui Kinzoku Completes Increase of Production Capacity for Ultra-Thin Electrodeposited Copper Foil with Carrier MicroThinTM, Technical Report, Malaysia, 2018.
[14] M. Kinzoku, Mitsui Kinzoku Completes Increase in Production Capacity of VSP® Electro-Deposited Copper Foil for High-Frequency Circuit Boards, Technical Report, Taiwan, 2019.
[15] A. Volvo, Volvo Group Annual Report, Technical Report, Sweden, 2007.
[16] M. Levin, Pharmaceutical process scale-up, CRC Press, 2005.
[17] S. Tsinontides, P. Rajniak, D. Pham, W. Hunke, J. Placek, S. Reynolds, Freeze drying—principles and practice for successful scale-up to manufacturing, International journal of pharmaceutics 280 (2004) 1–16.
[18] A. Faure, P. York, R. Rowe, Process control and scale-up of pharmaceutical wet granulation processes: a review, European Journal of Pharmaceutics and Biopharmaceutics 52 (2001) 269–277.
[19] H. Leuenberger, 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 (2001) 279–288.
[20] A. Tuchlenski, A. Beckmann, D. Reusch, R. Düssel, U. Weidlich, R. Janowsky, Reactive distillation—industrial applications, process design & scale-up, Chemical Engineering Science 56 (2001) 387–394.
[21] A. M. Deif, W. ElMaraghy, Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system, The International Journal of Advanced Manufacturing Technology 32 (2007) 557–562.
[22] G. Putnik, A. Sluga, H. ElMaraghy, R. Teti, Y. Koren, T. Tolio, B. Hon, Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap, CIRP Annals 62 (2013) 751–774.
[23] E. Fricke, A. P. Schulz, Design for changeability (dfc): Principles to enable changes in systems throughout their entire lifecycle, Systems Engineering 8 (2005) no–no.
[24] A. M. Ross, D. H. Rhodes, D. E. Hastings, Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value, Systems Engineering 11 (2008) 246–262.
[25] H. Almgren, Pilot production and manufacturing start-up: the case of volvo s80, International Journal of Production Research 38 (2000) 4577–4588.
[26] N. Lohse, Towards an ontology framework for the integrated design of modular assembly systems, Ph.D. thesis, University of Nottingham Nottingham, 2006.
[27] M. K. Chinnathai, B. Alkan, R. Harrison, Convertibility evaluation of automated assembly system designs for high variety production, Procedia CIRP 60 (2017) 74–79.
[28] Y. Koren, A. G. Ulsoy, Reconfigurable manufacturing system having a production capacity method for designing same and method for changing its production capacity, 2002. US Patent 6,349,237.
[29] B. Devlin, J. Gray, B. Laing, G. Spix, Scalability terminology: Farms, clones, partitions, packs, racs and raps, arXiv preprint cs/9912010 (1999).
[30] M. Slamanig, H. Winkler, An exploration of ramp-up strategies in the area of mass customisation, International Journal of Mass Customisation 4 (2011) 22–43.
[31] M. Wirges, A. Funke, P. Serno, K. Knop, P. Kleinebudde, 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 (2013) 57–64.
[32] J. Stauder, S. Buchholz, F. Klocke, P. Mattfeld, A new framework to evaluate the process capability of production technologies during production ramp-up, Procedia CIRP 20 (2014) 126–131.
[33] L. Surbier, G. Alpan, E. Blanco, A comparative study on production ramp-up: state-of-the-art and new challenges, Production Planning & Control 25 (2014) 1264–1286.
[34] G. Lanza, A. Sauer, Simulation of personnel requirements during production ramp-up, Production Engineering 6 (2012) 395–402.
[35] H. Almgren, Towards a framework for analyzing efficiency during start-up:: An empirical investigation of a swedish auto manufacturer, International journal of production economics 60 (1999) 79–86.
[36] P. D. Ball, S. Roberts, A. Natalicchio, C. Scorzafave, Modelling production ramp-up of engineering products, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 225 (2011) 959–971.
[37] S. C. Doltsinis, S. Ratchev, N. Lohse, A framework for performance measurement during production ramp-up of assembly stations, European journal of operational research 229 (2013) 85–94.
[38] S. Elstner, D. Krause, Methodical approach for consideration of ramp-up risks in the product development of complex products, Procedia CIRP 20 (2014) 20–25.
[39] O. Guschinskaya, A. Dolgui, N. Guschinsky, G. Levin, A heuristic multi-start decomposition approach for optimal design of serial machining lines, European Journal of Operational Research 189 (2008) 902–913.
[40] W. Wang, Y. Koren, Scalability planning for reconfigurable manufacturing systems, Journal of manufacturing systems 31 (2012) 83–91.
[41] A. Bensmaine, M. Dahane, L. Benyoucef, A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment, Computers & Industrial Engineering 66 (2013) 519–524.
[42] U. Ghani, System optimisation through the integration of virtual engineering and discrete event simulation, Ph.D. thesis, Loughborough University, 2013.
[43] G. Michalos, A. Fysikopoulos, S. Makris, D. Mourtzis, G. Chryssolouris, Multi criteria assembly line design and configuration–an automotive case study, CIRP Journal of Manufacturing Science and Technology 9 (2015) 69–87.
[44] P. Renna, Decision-making method of reconfigurable manufacturing systems' reconfiguration by a gale-shapley model, Journal of Manufacturing Systems 45 (2017) 149–158.
[45] M. Manzini, J. Unglert, D. Gyulai, M. Colledani, J. M. Jauregui-Becker, L. Monostori, M. Urgo, An integrated framework for design, management and operation of reconfigurable assembly systems, Omega 78 (2018) 69–84.
[46] A. Hafezalkotob, S. Chaharbaghi, T. M. Lakeh, Cooperative aggregate production planning: a game theory approach, Journal of Industrial Engineering International 15 (2019) 19–37.
[47] A. Kampker, K. Kreisköther, M. Hehl, S. Gillen, M. Rothe, Discrete event simulation approach considering scalable systems and non-expert users in the early phase of production planning for electric powertrains, in: 2017 6th International Conference on Industrial Technology and Management (ICITM), IEEE, pp. 139–144.
[48] B. R. Ferrer, B. Ahmad, D. Vera, A. Lobov, R. Harrison, J. L. M. Lastra, Product, process and resource model coupling for knowledge-driven assembly automation, at-Automatisierungstechnik 64 (2016) 231–243.
[49] P. G. Maropoulos, D. Ceglarek, Design verification and validation in product lifecycle, CIRP annals 59 (2010) 740–759.
[50] A. Caggiano, R. Teti, Digital factory technologies for robotic automation and enhanced manufacturing cell design, Cogent Engineering 5 (2018) 1426676.
[51] M. K. Chinnathai, Z. Al-Mowafy, B. Alkan, D. Vera, R. Harrison, A framework for pilot line scale-up using digital manufacturing, Procedia CIRP 81 (2019) 962–967.
[52] H. Knublauch, R. W. Fergerson, N. F. Noy, M. A. Musen, The protégé owl plugin: An open development environment for semantic web applications, in: International semantic web conference, Springer, pp. 229–243.
[53] V. Jain, M. Singh, Ontology development and query retrieval using protégé tool, International Journal of Intelligent Systems and Applications (IJISA) 5 (2013) 67–75.
[54] A. Ladj, Z. Wang, O. Meski, F. Belkadi, M. Ritou, C. Da Cunha, A knowledge-based digital shadow for machining industry in a digital twin perspective, Journal of Manufacturing Systems (2020).

[55] T. R. Gruber, et al., A translation approach to portable ontology specifications, Knowledge acquisition 5 (1993) 199–221.

[56] M. Uschold, M. Gruninger, Ontologies: Principles, methods and applications, The knowledge engineering review 11 (1996) 93–136.

[57] D. Penciuc, A. Durupt, F. Belkadi, B. Eynard, H. Rowson, Towards a plm interoperability for a collaborative design support system, Procedia Cirp 25 (2014) 369–376.

[58] N. Lohse, H. Hirani, S. Ratchev, Equipment ontology for modular reconfigurable assembly systems, International Journal of Flexible Manufacturing Systems 17 (2005) 301.

[59] M. Ahmad, B. Ahmad, R. Harrison, B. Alkan, D. Vera, J. Meredith, A. Bindel, A framework for automatically realizing assembly sequence changes in a virtual manufacturing environment, Procedia CIRP 50 (2016) 129–134.

[60] B. R. Ferrer, B. Ahmad, A. Lobov, D. Vera, J. L. M. Lastra, R. Harrison, 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.

[61] A. Negahban, J. S. Smith, Simulation for manufacturing system design and operation: Literature review and analysis, Journal of Manufacturing Systems 33 (2014) 241–261.

[62] A. Azab, T. AlGeddawy, et al., Simulation methods for changeable manufacturing, Procedia CIRP 3 (2012) 179–184.

[63] M. Jahangirian, T. Eldabi, A. Naseer, L. K. Stergioulas, T. Young, Simulation in manufacturing and business: A review, European Journal of Operational Research 203 (2010) 1–13.

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

[65] A. Konak, D. W. Coit, A. E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, Reliability Engineering & System Safety 91 (2006) 992–1007.

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

[67] A. Gosavi, et al., Simulation-based optimization, Springer, 2015.

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

[69] C. Nyulas, M. O'connor, S. Tu, Datamaster–a plug-in for importing schemas and data from relational databases into protege, in: 10th international Protégé conference, Citeseer, pp. 15–18.

[70] L. Zhang, J. Li, Automatic generation of ontology based on database, Journal of Computational Information Systems 7 (2011) 1148–1154.

[71] Y. Lu, X. Xu, A semantic web-based framework for service composition in a cloud manufacturing environment, Journal of Manufacturing Systems 42 (2017) 69 – 81.