



# A data-driven simulation to support remanufacturing operations

Paul Goodall\*, Richard Sharpe, Andrew West

Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, United Kingdom

## ARTICLE INFO

### Article history:

Received 12 June 2018

Received in revised form 5 October 2018

Accepted 9 November 2018

Available online 24 November 2018

### Keywords:

Data-driven simulation

Remanufacturing

Industry 4.0

Discrete event simulation

Operations management

## ABSTRACT

Simulations are a vital component in developing smart manufacturing systems, predicting the behaviour of the manufacturing shop floor operations to support production planning, scheduling and maintenance decisions within manufacturing environments. However, simulations are often limited in their ability to support real-time business decisions in complex fast changing environments due to the cost and time required to build, update and maintain simulation models. Remanufacturing operations in particular could benefit from the use of simulations as a tool to support the assessment of different strategies to real-time scenarios due to the uncertain nature of product returns.

This research develops a data-driven simulation approach to predict material flow behaviour within remanufacturing operations, by utilising data from digital manufacturing systems (i.e. databases, traceability systems, process plans) to update and automatically modify the simulation constructs to reflect the real world or planned system. A data-driven simulation is proposed comprising of three elements: (i) an *adaptive remanufacturing simulation* algorithm to model the complex material flow found within a remanufacturing process in a generic and reusable way, (ii) a *remanufacturing information model* to structure and highlight the simulation data requirements and (iii) an *information service layer* to collect and analyse sensor data for use within the simulation. The simulation is implemented to demonstrate how it can automatically reconfigure and adapt to changes within the data inputs (process and factory models) using a case study of operations from a Waste Electrical and Electronic Equipment (WEEE) remanufacturer, utilising data collected from a Radio Frequency Identification (RFID) traceability system installed within the remanufacturing facility.

© 2018 Published by Elsevier B.V.

## 1. Introduction

Simulation techniques, such as Discrete Event Simulations (DES), System Dynamics (SD) and Agent-Based (AB) simulations, are commonly used to predict the behaviour of real complex manufacturing systems [1]. These simulation techniques have been used to support manufacturing decisions (highlighted in Fig. 1), including designing factory layouts, supply chain interoperations and developing production planning strategies [1–4]. However, simulations are often developed as a stand-alone application, requiring data and business rules to be manually embedded within them by domain and simulation experts, which means they can be expensive and time consuming to build and maintain, thus limiting their use to high value strategic decisions such as factory design rather than day to day operations management. As real manufacturing systems are subject to constant change, due to factors such as machine wear and failure, demand

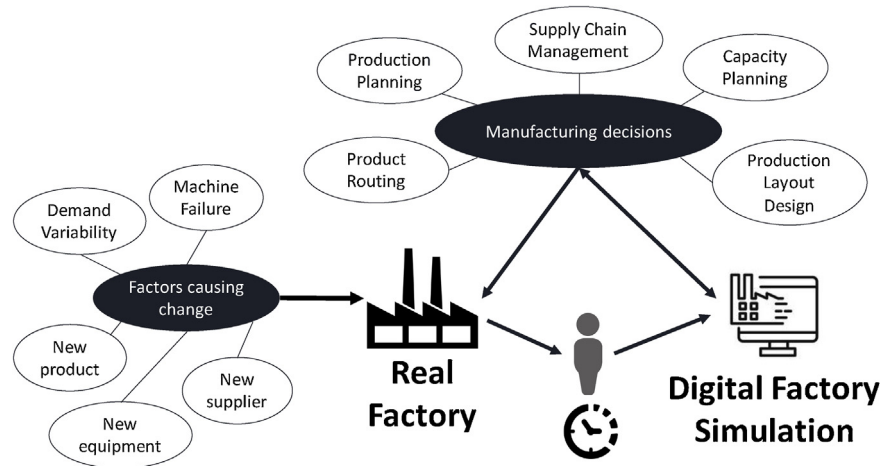
variability and the introduction of new products and equipment, simulations can quickly become outdated [5]. The manual task of reconfiguring a simulation to accommodate real-world changes limits the use of simulations to single-use applications and has led to simulation activities gaining the reputation as a “throw away” tool [6–9].

To support dynamic planning and management tasks better, simulations must be able to adapt to reflect the real world and its changing environment automatically by utilising existing digital artefacts, such as ad-hoc databases, documents and formal systems (e.g. Enterprise Resource Planning (ERP)), to provide information and modify the simulation constructs automatically, enabling non-specialists to benefit from the evaluation processes [10,11].

Data-driven simulation is an approach where the simulation is completely parameterised by providing data through a set of data inputs, allowing a user to create and run a simulation model without the need to do any programming [10]. Whilst data-driven simulations have been developed for domains including automotive manufacturing [12,13], assembly logistics [14,7,10], semiconductor manufacturing [15] and building construction [16], there

\* Corresponding author.

E-mail address: [P.A.Goodall@lboro.ac.uk](mailto:P.A.Goodall@lboro.ac.uk) (P. Goodall).



**Fig. 1.** Schematic of the current simulation process, highlighting the reliance upon humans to update information between real factory and simulation manually.

has been little attention given to area of End of Life (EoL) operations such as remanufacturing. Remanufacturing operations, where used and discarded products are processed to restore them to like-new conditions, exhibit complex and variable material routing behaviour due to high levels of uncertainty surrounding the products that are processed (i.e. type, condition, availability, value).

Remanufacturing operations could benefit from the use of simulations to support dynamic planning and scheduling, however, there is currently no consensus upon how to develop data-driven simulations to ensure they model remanufacturing characteristics and can self-adapt to user and system changes.

The aim of the research outlined in this paper is to develop a data-driven simulation suitable for modelling the complexities of remanufacturing operations. The paper has been organised as follows:

- Section 2 identifies the related work regarding the challenges of remanufacturing operations and data-driven simulations.
- The design of the reconfigurable simulation for remanufacturing is outlined in Section 3 and described in detail within Section 4.
- Section 5 implements and demonstrates the simulation using a case study for a Waste Electronic and Electrical Equipment (WEEE) processor.

## 2. Related literature

### 2.1. Remanufacturing

Remanufacturing enables the reuse of used and waste products (often referred to as *cores* [17]) through a series of repair and restoration activities to return them to “like new conditions” [18,19]. Successful remanufacturing provides an economical method of supporting aftermarket activities, such as warranty returns and maintaining a supply of spare parts and components [20], while refurbishment and repair activities can offer low-cost product alternatives for secondary markets [21]. The environmental impact of these activities is lessened due to the recovery of both the material content and embedded energy from manufacturing, leading remanufacturing to be seen as a key component in developing a truly *circular economy* [22]. Whilst remanufacturing operations face similar challenges to traditional manufacturing (i.e. Inventory management and Production Scheduling), they are complicated due to high levels of uncertainty caused by variable and uncertain cores characteristics [23].

Uncertainty within remanufacturing is caused by limited information, such as the type, condition and quantity of cores, flowing from the operational phase of the products life cycle to the remanufacturer [23–26]. This uncertainty causes additional remanufacturing processing in the form of inspections and disassembly to determine a cores suitability for remanufacture [27,28]. Varying degrees of core condition, identified during physical inspections, means that not every operation is required to remanufacture a core, resulting in potentially unique process routing for each part [29,30]. Not all cores are suitable for remanufacturing due to variable market values [31,32], the condition of the core [33,31] and other business and legislative constraints (e.g. the requirement in the UK for all data holding devices to be securely erased in adherence to Publicly Available Specification (PAS) 141:2011 [34]), resulting in alternative EoL strategies (i.e. refurbishment, parts cannibalisation, recycling and disposal) being selected [35]. Further challenges include the need to disassemble cores and later reassemble the remanufactured or replaced components, adding further complexity within the material flow [23].

Digital modelling and simulation could be a useful tool to support decision making within remanufacturing operations to assess the benefits and risks of strategies for production scheduling, planning and other shop floor decisions faced with high levels of uncertainty and complexity. Several researchers have utilised simulation of remanufacturing operations to investigate the effect of uncertainty within production parameters such as the arrival timing, quantity and quality of cores, using probabilistic distributions [17,36–38]. However, the simulation models developed are often simplified versions of real-world remanufacturing operations and rely upon static user input of data collected from interviews and historical observations of the shop floor [17]. This limits the use of simulations as a tool for high-level strategic decisions, rather than detailed tactical and operational decision making, where accurate representations of the real system need to be continually updated to reflect the fast-changing market conditions [33]. Integration of data from various digital business sources to update and adapt a simulation model automatically would be a useful tool to support remanufacturing operational decisions.

### 2.2. Data-driven simulation

Developing methods that reduce the complexity to build and maintain simulations has been the focus of some researchers with two approaches often used: *modular simulation* and *data-driven simulation*.

*Modular simulations* reduce the complexity of developing simulations by creating generic templated modules which can be reused to model the behaviour of a particular domain element using specific simulation constructs, usually configured through a Graphical User Interface (GUI). Many commercial simulation packages offer this functionality such as (e.g. Simulink, Anylogic, Witness) for common manufacturing operations, whilst researchers have looked to extend these capabilities for more specific domain scenarios such as remanufacturing [39], electronics manufacturing [40] and Just in Time (JIT) manufacturing [41]. However, whilst this method can support the creation of the simulation it still requires a user to embed the specific business rules (i.e. process routing and historical data) into the simulation manually, limiting the maintainability and adaptability to changes in the real world.

On the other hand, data-driven simulations are models that can be completely parameterised so data can be input and adjusted externally from the simulation, allowing a user to create, run and update simulation model without the need to do any programming [10]. The degree of parameterisation within data-driven simulation in research varies, affecting the flexibility of scenarios which can be modelled. Three types of categories of data-driven simulation have been identified within the literature: *historical process data*, *factory status information* and *automated model generation*.

*Historical process data* utilises data recorded from shop floor activities, such as Manufacturing Execution Systems (MES's) [13], to input data into a simulation rather than rely upon assumptions or manually timed processes. Using *historical process data* can reduce the inaccuracies within the input data which may quickly become outdated and inaccurate, whilst also reducing the effort of collecting and inputting data into simulations. This information is used within manufacturing simulations to determine processing timings and decision probabilities as well as the variation within these outputs.

*Factory information* relates to the current status of factory elements such as objects, resources, plans and schedules and is often referred to as “real-time” information. This information can be obtained from real production systems to reflect the current status of the factory or can be adjusted to perform “what-if” analysis, either manually or automatically as part of an optimisation algorithm. Examples of *factory information* use within data-driven simulation include the status of machines [13], resources (i.e. worker shift) [13], production schedules, inventory and WIP in the factory [12].

*Automated model generation* enables automatic reconfiguration of the simulation components (i.e. queues, delays and their routing [42]) to reflect changes within input parameters, such as Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) files, rather than embedding the business rules directly into the simulation [43]. This approach emphasises the reuse of simulations within the particular domain, enabling greater flexibility within the simulations to reflect the change within the real or envisaged system. Automated model generation is particularly useful when modelling complex processing logic or when the

domain are subject to frequent changes (such as flexible manufacturing), easing the maintenance of the simulation to real-world changes.

A summary of data-driven simulations developed within published research is presented within Table 1. Much of the research developed has focused upon *historical data* and *factory status information* as the key data inputs. Whilst these approaches cater for manufacturing systems where changes in the process seldom occur, thus requiring no restructuring of the core simulation blocks (i.e. the fundamental simulation elements), limited research has been conducted for *automated model generation* which would benefit more complex manufacturing scenarios where process rules are numerous and highly flexible, such as remanufacturing.

While there are many papers detailing models and simulation for remanufacturing activities in literature [3], most model remanufacturing at a high level with very few detailing specific remanufacturing operations, of which even fewer adopt a data-driven approach. Franke et al. [37] use an adaptive discrete event simulation which changes within planning of capacities and production programs for mobile phone remanufacture based upon an Integer Linear Programming (ILP). The simulation is based upon a generic view of the remanufacturing process for a mobile phone, whilst facility resource features, such as workers and conveyors are generated at runtime based upon the ILP requirement. Whilst this is suitable for the specific example of mobile phone remanufacture, it is not possible to extend or modify the process (such as adding extra routes for different product type, e.g. a laptop, server, disk drive) without changing the hard-coded simulation. For companies focused on the business of WEEE, this limits the usefulness of the simulations to support the range of business decisions that they need to remain competitive.

Wy et al. [7] and Wang et al. [10] have developed process adaptable data-driven simulations for the assembly logistics domain. Both have used a similar approach to abstract the logistics process into its fundamental components, leading to the development of a generic model. This generic model provides an abstracted view of the particular process, which is tailored to the specific scenario through input parameters which are outlined through an information model and with data provided via a set of services. However, remanufacturing contains specific processes (i.e. assembly, disassembly, inspections) whose behaviours aren't represented within these simulation architectures. Within this research, the work Wy et al. [7] and Wang et al. [10] is expanded by developing a data-driven simulation approach for modelling remanufacturing operations.

### 3. Overview of the data-driven remanufacturing simulation

An overview of the data-driven remanufacturing simulation is presented within this section. The simulation is a self-contained service which users can interact with via interfaces to set simulation parameters and connect with external data sources. Within the

**Table 1**  
Data-driven simulations within production environments.

	Application Domain	Historical Process Data	Factory Status Information	Automated model generation
Frantzen et al. [12]	Automotive manufacturing	✓	✓	
Ghani et al. [13]	Automotive manufacturing	✓	✓	
Wy et al. [7]	Assembly logistics		✓	✓
Wang et al. [10]	Assembly logistics	✓	✓	✓
Tavakoli et al. [14]	Assembly logistics	✓	✓	
Bagchi et al. [15]	Semiconductor manufacturing	✓	✓	
Azimi and Lee [16]	Building construction	✓	✓	
Franke et al. [37]	WEEE/EoL Recovery	✓	✓	

simulation itself there are three core elements (i.e. the adaptive remanufacturing simulation, remanufacturing information model and an information service layer) as shown in Fig. 2 and described below.

The **Adaptive Remanufacturing Simulation**, described in detail in Section 4.1, automatically creates and adapts the simulation to a particular remanufacturing scenario using real-world data captured within an organisation. A generic model of the remanufacturing material flow is developed, providing both a structure to organise DES elements (presented in Fig. 4) and a set of procedures (shown in Table 2) to request information from the Remanufacturing Information Model and the Information Service Layer. Adaptation of the simulation to the particular remanufacturing scenario occurs both structurally, through the automatic addition of extra DES elements to perform remanufacturing activities and with the behaviour of cores based upon the response of the procedures (i.e. processes, time spent in an activity, inspection pass probability). This approach removes the scenario specific logic away from the simulation programming and into the data input parameters, enabling the reuse of the simulation and automatic simulation generation and is based upon the work of Wy et al. [7] who use a similar approach develop a data-driven simulation for logistics-embedded assembly manufacturing lines.

The **Remanufacturing Information Model** provides a data structure for modelling real-world remanufacturing operations in a generic manner. It contains three elements: (i) the *core* which is to be remanufactured, (ii) the *factory and resources* which form the factory environment including staff and machines, WIP and schedules and (iii) the *process* describing the interaction of *cores*

and *resources*. This architecture is based on the generic product, process, resource view for describing manufacturing activities as it enables the key elements of the factory to be described and relationships with each other [44]. The specific data requirements for each of these elements are detailed in Section 4.2.

The **Information Service Layer**, described within Section 4.3, stores reusable services to retrieve and transform data, which are collected from factories and the supply chain (such as traceability records), into information to be used within the simulation. Examples of these services include determining statistical distributions for activity times, pass and failure rates from the results of real-world inspections and current resource prices. Interoperability between the services and the simulation has been achieved by adopting a Service Oriented Architecture (SOA) approach, which enables services to be designed and deployed in the developer's language(s) of choice (e.g. C#, SQL, PHP, Python).

#### 4. Design of the reconfigurable remanufacturing simulation

##### 4.1. Adaptive remanufacturing simulation

To ensure reusability and reduce the time and cost of maintenance, the simulation should adapt to changes in the data and information describing the remanufacturing scenario. The adaptive remanufacturing simulation is based upon a generic view of material flow in remanufacturing operations where a core can be in one of two states: *waiting* or *processing* as shown in Fig. 3. Cores enter the remanufacturing facility and are set to a waiting state

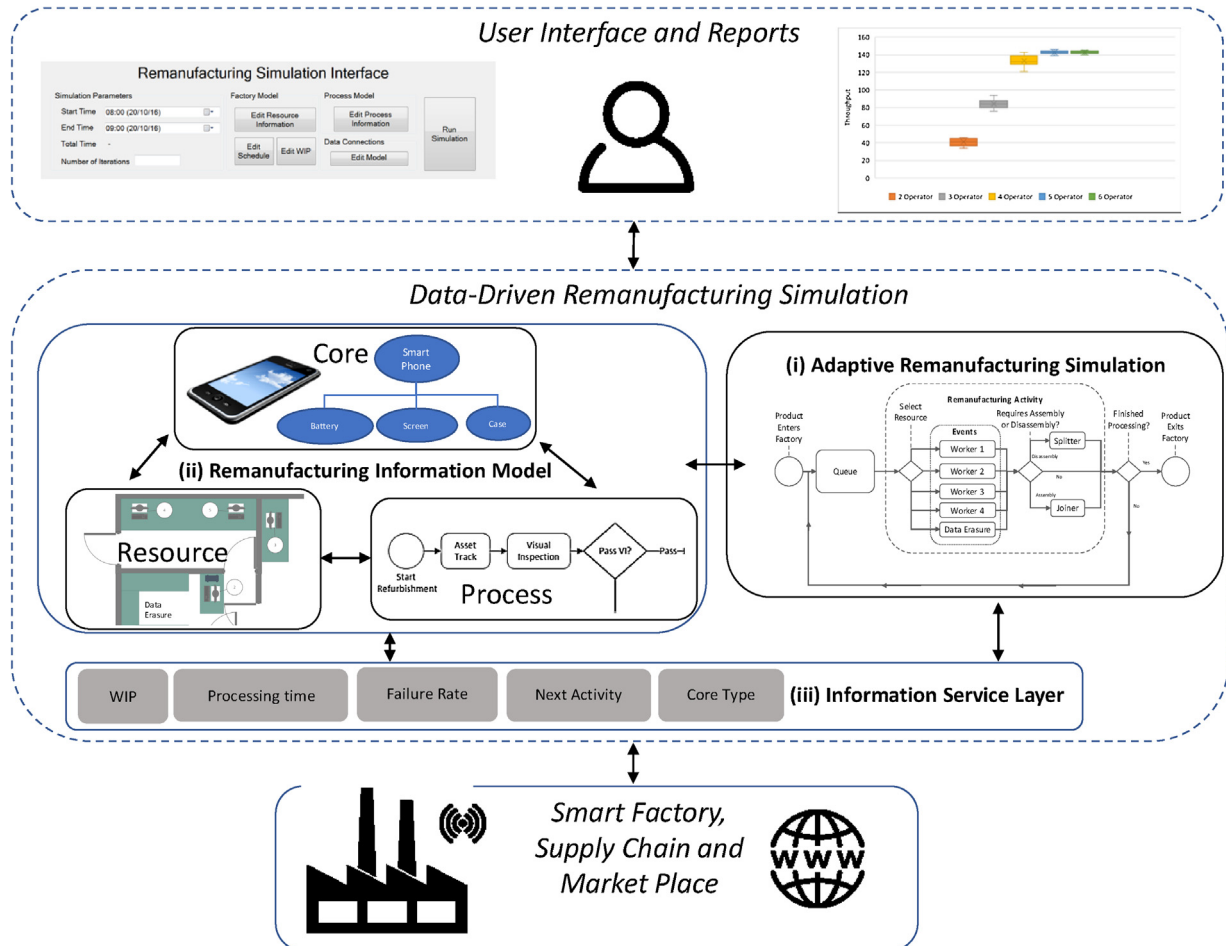


Fig. 2. Overview of the simulation frameworks three aspects: Adaptive Generic Remanufacturing Simulation, Remanufacturing Information Model and the Information Service Layer.

**Table 2**  
Services for generic model.

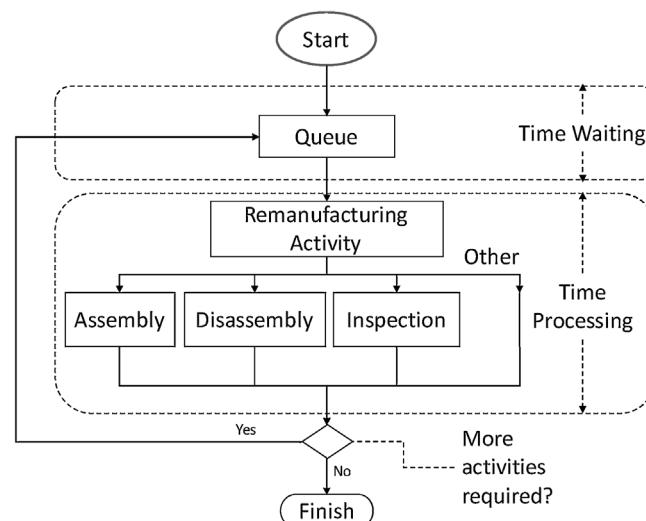
Function Name	When does it occur	Description
Generate_WIP	Prior to simulation start	Preloads the simulation model with the current WIP in the factory based upon sensor readings and sends them to the queue.
Generate_Cores_Info	Emitter after generation	Stochastically assigns core information based upon historical data (i.e. type, manufacturer, model) for new cores entering the simulation.
Set next arrival time	Emitter after generation	Sets the time for next set of cores to be generated into the simulation. This can be based upon known batch arrivals or random intervals using historical data trends.
Determine Resource Requirements	Upon entering queue	Determines the resources required for the next activity to ensure that the core is processed by a suitable resource.
Set_Priority	Entering queue	Set the priority of the product to be reprocessed. This can enable certain weightings to be assigned.
Conduct_Process	Upon entity entering resource	Determines the time taken for an activity to be performed by a resource on a product. This is stochastically derived based upon historical data.
Inspection	Upon entity leaving resource	Stochastically assigns a value to an attribute based upon historical data.
Disassemble	Entering splitter	Determines the number and type of components to disassemble.
Assemble	Entering joiner	Determines the components required for assembly the product.
Determine_next_activity	After entity leaves resource	Determines which activity is required next by consulting the process model.

until the resources required to perform an activity (such as a human operator) become available. During processing, cores are held by single or multiple resources while activities (i.e. disassembly, assembly, rework, inspections) are performed. The type of activity undertaken impacts on the material flow. For example, disassembly and assembly operations can increase or reduce the number of parts which are independently flowing through the factory. Inspections will generate additional information which affects the future activities applied to the core, for example, whether the core passes or fails any test criteria associated with quality requirements or required functionality. After *processing*, a decision is taken to determine if further *processing* is required. If so the core returns to a *waiting* state, and if not, it exits the remanufacturing process.

This generic view of remanufacturing operations has been transformed into a DES using common simulation blocks, shown in Fig. 4. Cores are represented within the simulation as *entities* which move between the *common simulation blocks*. The *emitter block* inputs new cores into the simulation for remanufacturing, either at the beginning of the simulation to represent the Work in Process (WIP) or at intervals during the simulation run representing new core arrivals. After input, cores move to the *queue* where they wait for a suitable resource to become available. A custom set of resources is generated automatically when the simulation is

initialised, based upon the number and type specified within the resource data model for the remanufacturing facility, ensuring the user doesn't need to modify the simulation manually. Individual resources are represented within the simulation by *delays*, which hold an entity for a specified period to signify an activity being performed. If the activity performed is assembly or disassembly, then the *joiner* or *splitter* blocks are used to combine or decompose entities respectively. As assembly and disassembly tasks are treated as a single activity, joining of cores occurs before the delay, whilst splitting occurs after, to avoid repeated occurrences of the activity due to additional entities which represent the sub-components. Once the remanufacturing activity has finished, the entity will then either flow back to the queue if further processes are required or exit the simulation via the *terminator*.

Information services are used to customise the response of the simulation for a particular remanufacturing business, listed in Table 2. These services are triggered when entities reach specific stages of the simulation, such as entering the queue and are used to determine the simulation behaviour such as the next activity in the process, the time an activity takes and the generation of inspection results. The results of these services are based upon information captured on the shop floor (e.g. see Fig. 8), or through process and resource models (e.g. see Fig. 10) which are created before initiation of the simulation.



**Fig. 3.** Abstracted view of remanufacturing operations, where the time a core spends at the remanufacturing facility can be viewed either as waiting or being processed.



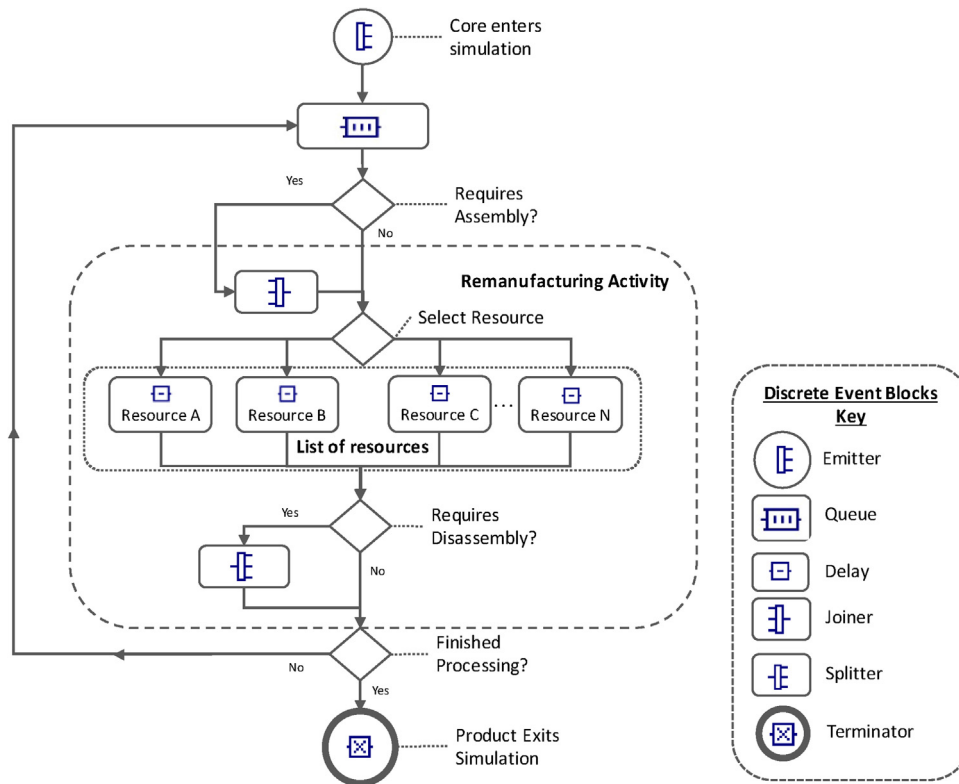


Fig. 4. The generic layout of discrete event simulation objects to represent the abstracted material flow through a remanufacturing process.

## 4.2. Remanufacturing information model

### 4.2.1. Core model

A generic data model for cores has been derived to enable essential simulation information to be represented, shown in Fig. 5, where each entity within the DES is assigned. The core data model has been influenced by research from the PROMISE project [45] and Um et al. [46] who have developed generic data models

for product life cycle management and EoL product recovery respectively. These approaches have been used as a starting point for the proposed data model but have been simplified based upon the needs of the simulation. The cores information model has been structured as follows: (i) *specification*, (ii) *structure*, (iii) *history* and (iv) *current status*.

The *Specification* section contains information to represent the core being remanufactured using two classes, *Identification*

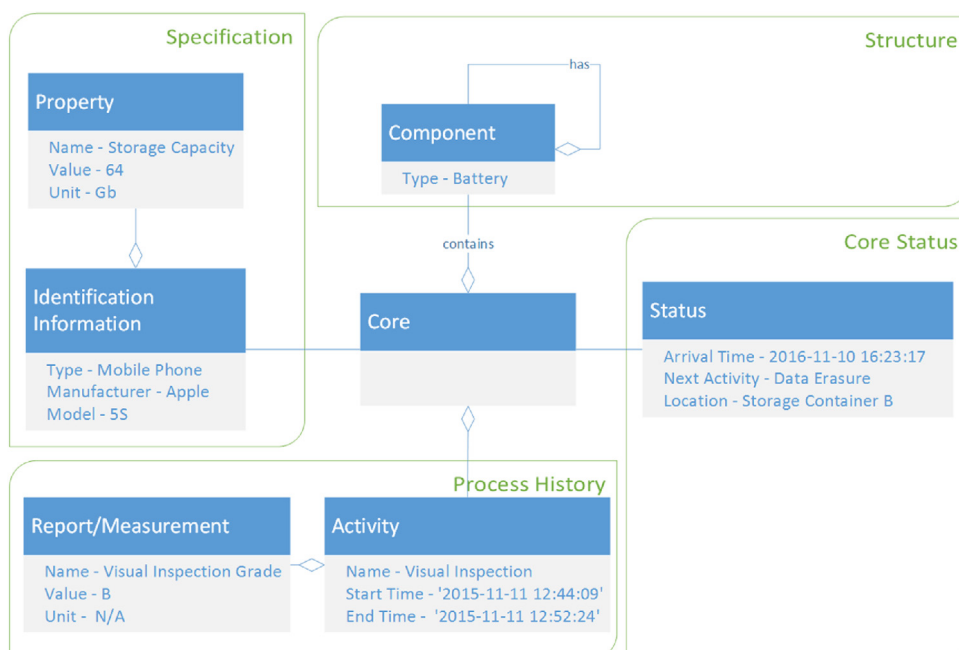


Fig. 5. Core Model with mobile phone example displayed using Unified Modelling Language (UML) Class diagram.

**Information and Property.** The *Identification Information* class contains generic information about a core such as a type (e.g. mobile phone, hard drive, laptop), manufacturer and model. The *Property* class provides a generic template to specify particular property attributes using the properties *name* (e.g. Hard Drive Size), *unit* (e.g. Gigabytes (Gb)) and *value* (e.g. 500). This information is used within the simulation to retrieve core specific information to enable a more customised estimate for activity times or inspections results.

The *Structure* describes the cores' assembly, such as the number and type of components and subassemblies. The *Component* class represents individual parts (e.g. batteries, material components, screens) which are specified using the *Type* attribute. Multiple instances of the component class can be used to denote more than one part, while subassemblies can comprise additional instances when further granularity is required to depict their constituent components. The information is used within the simulation during disassembly and assembly activities, to determine how many further entities to add or combine.

The *Process History* section records the outcomes of any simulated activities, inspections or measurements to determine future routing decisions required within the simulation and comprises two classes: *Activity* and *Measurement*. The *Activity* class records general information about the activity conducted, such as the type/name, the start and end time as well as the resources which were used. The *Measurement* class records the result of an activity (i.e. if an inspection passed or failed) in a generic manner such as the property class within the Specification section.

The core *Status* represents the current status of the core within the simulation. This is used to store information such as the next activity to be performed and current location within the factory (Note: location information has not been the focus of this research. However, it will be reported in future work when reporting the results of modelling the logistical issues associated with remanufacturing supply chains).

#### 4.2.2. Factory and resource model

The remanufacturing facility is modelled within the simulation architecture using the data structure shown in Fig. 6. The model comprises: (i) *Resources* to perform remanufacturing operations, (ii) *Work in Progress (WIP)* and (iii) *Production Planning* information.

Resources are split into two categories, *active* and *passive*. *Active* resources are those which perform an activity such as a machine or human worker. Within the DES, each *active* resource is modelled as a delay (shown in Fig. 4). *Passive* resources are used by *active* resources to perform activities, such as materials (e.g. metals), tools (e.g. drill, mills) energy and water. *Passive* resources can constrain the process if they are not available when required and their consumption incurs costs.

*WIP* lists cores which are being currently being processed within the facility. Each core has an information structure outlined in Section 4.2.1, therefore expressing previous activities and inspection results. This information is gathered using the *Generate\_WIP* function (described in Table 2) prior to running the simulation and is used to preload the simulation with the current *WIP*.

The *Production Planning* section contains management information regarding the resource schedule (i.e. staffing plan for the simulation horizon and any scheduled maintenance) and information about planned batches such as the expected time of arrival and quantity.

#### 4.2.3. Process model

The *Process* model describes the steps, activities, decisions and resources required to remanufacture a core. Due to the uncertain nature of a returned core, the process may contain paths for other levels of recovery (i.e. remanufacture, refurbish, repair, recycle, disposal) which are considered during the process. During simulation, the *Process* model is interrogated to determine the next activity (if any) to perform, based on the results of past activities and inspections, and the resources are required.

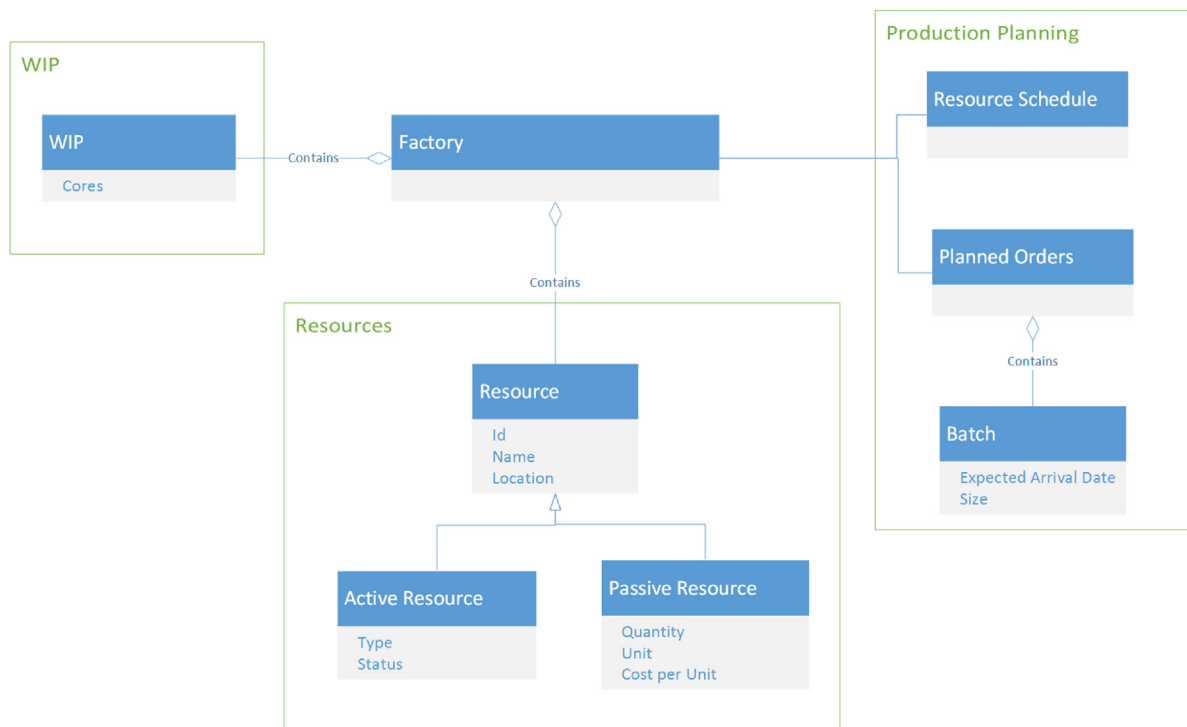


Fig. 6. Factory Data Structure using UML class diagram notation.

The data structure used to represent the remanufacturing process is outlined in Fig. 7. The data structure utilises the Business Process Modelling Notation (BPMN) [47] due to its ability to model complex workflows and decisions in supporting process flow representations.

The remanufacturing process is described using the primary process elements: *Start*, *Finish*, *Activity*, *Gateway* and *Sequence*. The *Start* class signifies the beginning of a process, while *Finish* class identifies the end and the EoL outcome (i.e. remanufacture, repair, recycle, disposal). Multiple *Finish* instances can be used within a process if multiple EoL outcomes arise, as can be seen in the process model within Fig. 10. The *Activity* class represents individual activities performed during a process, such as an assembly, disassembly, repair, inspection and cleaning. The *activity* class has been extended for *Assembly*, *Disassembly* and *Inspection* to include additional attributes required to perform simulation, such as a list of disassembled components and links to the services (located in the *Information Service Layer*) used to generate information from the inspection. Resources are needed to perform each activity and are represented within the *Resource Requirements* class which specifies the resource type and quantity.

The *Gateway* class is used to direct the material flow at junctions within the process, such as determining the appropriate activity to perform after inspection. Each potential route following a decision is represented as an *Outgoing Route* class which is a type of *Sequence* that is used to link process blocks together to form a process sequence. The rules dictating the routing of decisions are represented within the *Condition* property of the *Outgoing Route*, using a property, operator and a value, which are assessed like "if-then-else" statements. For example, if the result of an inspection is

deemed to have passed the condition would resemble "Property='Passed'".

#### 4.3. Information service layer

Central to the successful deployment of the simulations within a business is the ability to input and update large amounts of data automatically to represent the real-world behaviour of the complex remanufacturing system in real-time. With Big Data analytics opportunities within the manufacturing domain arising through the deployment of technologies such as Radio Frequency IDentification (RFID) and Industry 4.0 embedded sensor systems, the challenges are to develop services to convert these raw data into useful information required to control the simulation.

An SOA approach has been adopted to enable integration of data between the source, information service and the simulation, depicted in Fig. 8. Services can transform data captured within the factory and the supply chain (e.g. traceability events, product locations, market prices) into information required to perform the simulation (highlighted in Table 2). These services are self-contained software components which can be written in an independent programming language to the simulation (i.e. C#, SQL, Php, Python), allowing reuse within different applications. The simulation communicates and requests information with these services using a RESTful Application Programming Interface (API). An example is depicted in Fig. 8 of the software component providing the service which retrieves the WIP from the factory database written in the SQL language.

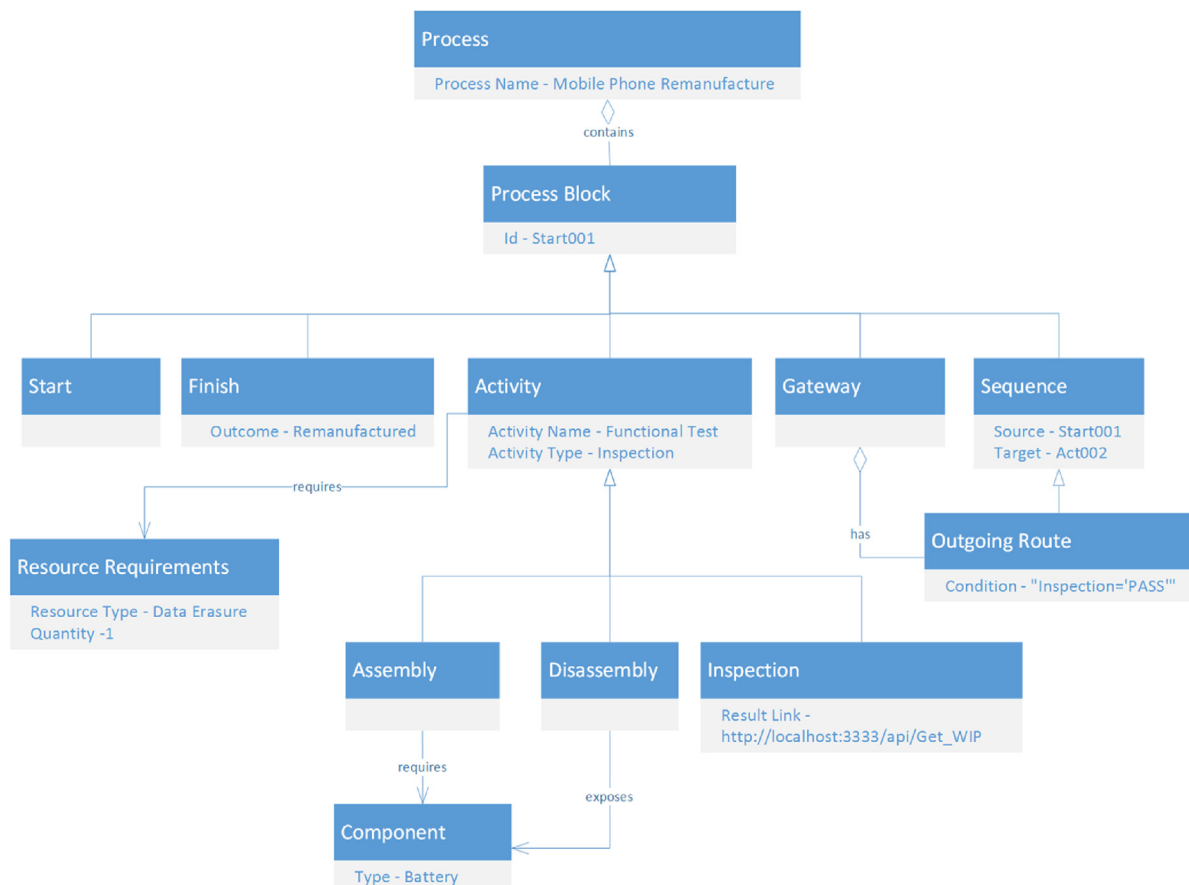
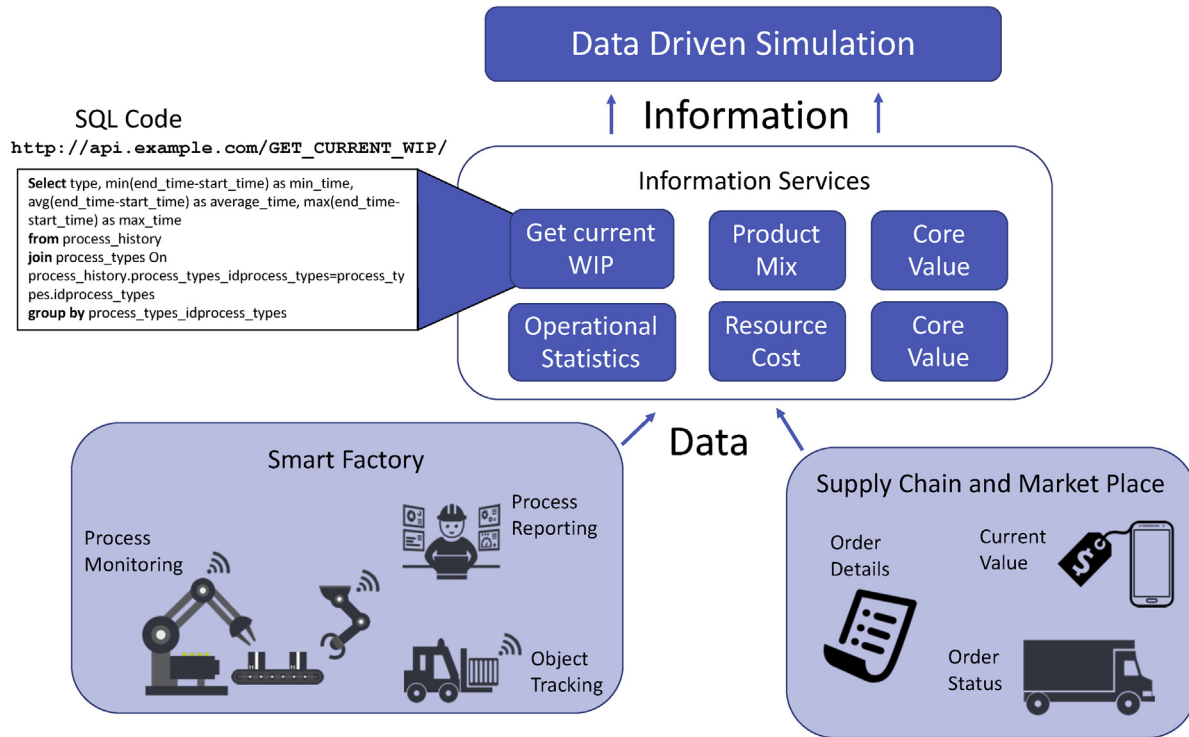


Fig. 7. Process Data Structure using UML class diagram notation.





**Fig. 8.** Schematic diagram depicting the information services transforming data captured throughout the supply chain, factory and market place into information to be used by simulation.

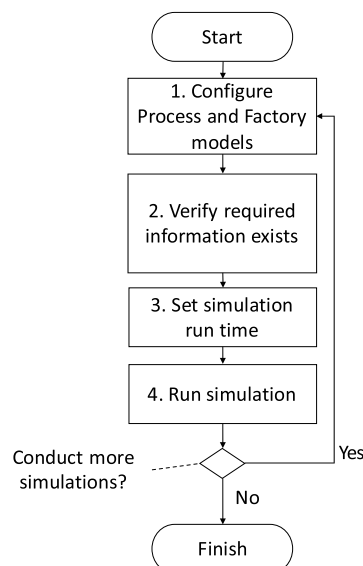
## 5. Simulation verification

This section demonstrates the implementation of the simulation framework using a case study example of a small UK-based independent business (<25 employees) which securely refurbishes WEEE containing sensitive data, such as mobile phones, laptops and hard drives for businesses and public organisations. The business uses an RFID traceability system [48] to individually track the progress of each core processed within the factory, enabling the analysis of historical data and real time monitoring of the factory status related to the current WIP. The data-driven simulation described in this paper is implemented for this case study, utilising

the traceability system as a data source. Implementation of the system is described within Section 5.1, whilst the simulation functionality is demonstrated with use cases in Section 5.2.

### 5.1. Simulation implementation

The remanufacturing simulation framework is implemented within C# programming language using the commercially available DES library DELSI [49]. DELSI was chosen as it enabled an object-oriented programming approach to assign additional class structures to simulation components (i.e. the information model structures described in 4.2), import data from external



**Fig. 9.** Operational procedure for the simulation.

databases, services and third-party systems, and dynamically control process routing through custom built algorithms rather than manual coding. DES objects (i.e. *emitter*, *queue*, *delays*, *splitters*, *terminators*) were structured to follow the structure presented in Fig. 4, while functions were built to implement the procedures described in Table 2. The information model described within Section 4.2 has been implemented using C# classes. The operational procedure for using the simulation is depicted within a flowchart in Fig. 9.

The first step is to configure the process and factory models to reflect the real system or the modified scenario to be investigated using the simulation. The remanufacturing process model is created using Camunda Modeller [50], an open source tool to build process models using the BPMN standard using a Graphical User Interface (GUI) which are then exported to the simulation using

eXtensible Markup Language (XML) format and parsed into an instance of the process model class. The resource model for the factory was created using a custom-built interface to enable resources to be added or removed before each simulation run. Examples of these models and the GUI inputs are shown in Fig. 10.

Shop floor operational data of individual processing activities, such as processing times, pass-fail rates and the product involved are collected using a traceability system and stored within a relational database. These data are analysed on-demand through bespoke REST web services, created using SQL language, to provide information such as activity times and pass-failure rates for particular products of interest, with the API's shown in Table 3. A consistency check is performed before simulating to ensure information about all processes and resources listed within the models are present. Any additional information (i.e. predictions for

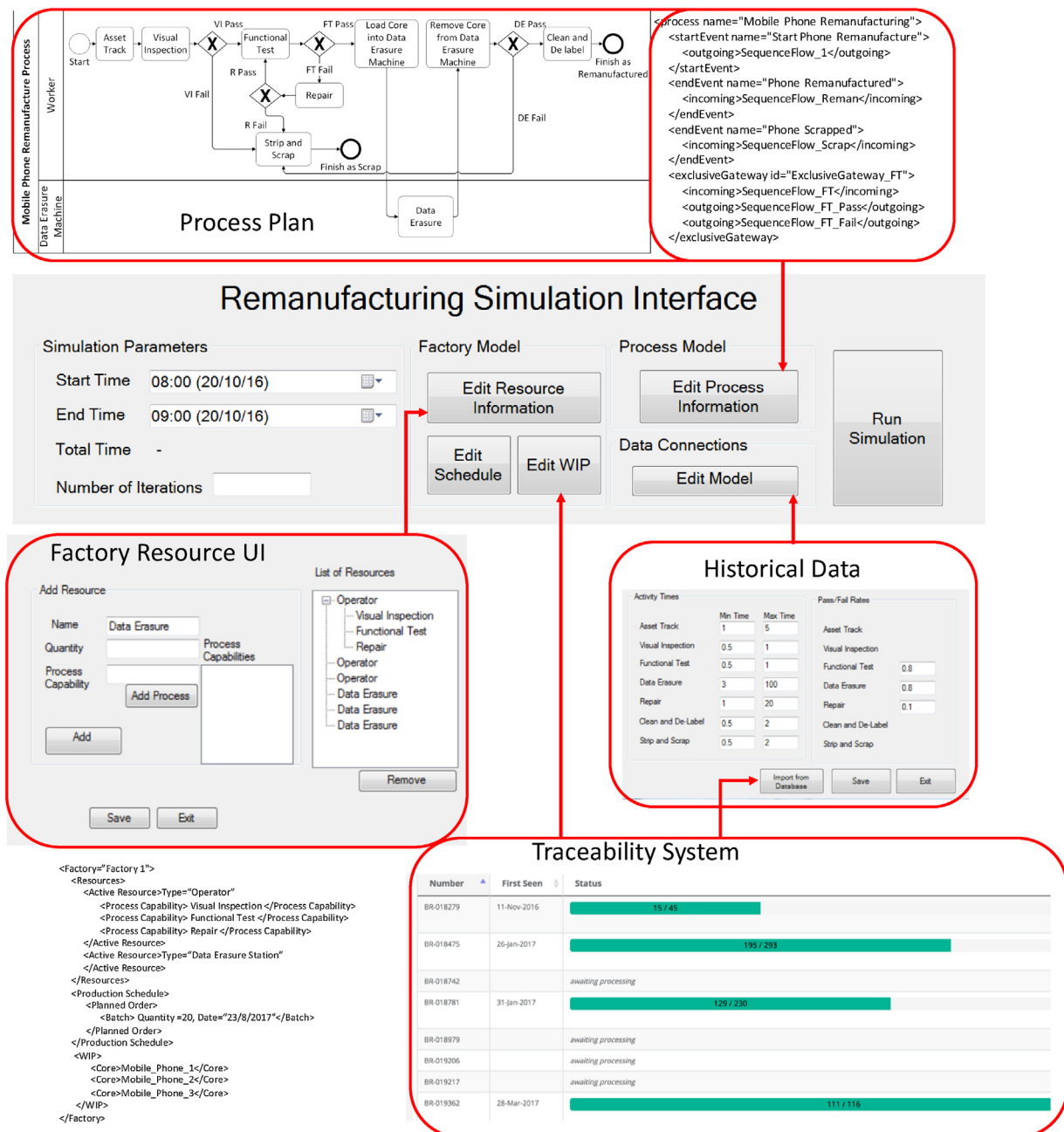


Fig. 10. Input of data into the simulation via user interfaces and external systems which are converted to the data structures proposed in Section 4.2 via XML shown with pseudo code examples.

**Table 3**

Web services to retrieve simulation information.

Function Name	URI	Request Type	Input Parameters
<i>Get_WIP</i>	<a href="http://localhost:3333/api/Get_WIP">http://localhost:3333/api/Get_WIP</a>	GET	N/A
<i>Generate_Cores_Info</i>	<a href="http://localhost:3333/api/Generate_Core_Info">http://localhost:3333/api/Generate_Core_Info</a>	GET	N/A
<i>Set next arrival time</i>	<a href="http://localhost:3333/api/Set_nx_arr">http://localhost:3333/api/Set_nx_arr</a>	GET	N/A
<i>Conduct_Process</i>	<a href="http://localhost:3333/api/Get_Activity_time?Activity=[activity]">http://localhost:3333/api/Get_Activity_time?Activity=[activity]</a>	GET	Activity Name
<i>Inspection</i>	<a href="http://localhost:3333/api/Get_Activity_insp?Activity=[activity]">http://localhost:3333/api/Get_Activity_insp?Activity=[activity]</a>	GET	Activity Name

activities yet to be implemented) are flagged and can be entered manually within the service via a custom built GUI.

Start and end times for the simulation are selected and the simulation is executed. After the simulation is complete, reports are generated informing the user upon the effects of key performance indicators, such as factory throughput and resource utilisation. Repeating the simulation through multiple iterations can be conducted to obtain mean and standard deviation of the above metrics, on which the uncertainty of the results can be measured.

## 5.2. Simulation demonstration

Three examples shown below highlight the flexibility of the simulation and its ability to automatically adapt to changes within the data inputs, without changing the simulation constructs. These examples reflect changes to: (i) resources, (ii) WIP, (iii) process plans.

### 5.2.1. Resources

The first change within the system is the number of resources used within the simulation. This is useful to see the effect of staffing levels on factory performance, which may vary due to operator illness, breakdown of machinery or the addition of new equipment. A simple scenario is demonstrated showing the effect of different staffing levels within the production facility highlighted within Fig. 10. Operators perform the majority of refurbishment operations within the process plan, shown at the top of Fig. 10. Simulation is performed with 2–6 operators to determine the required number complete the processing of 150 cores in a single day. Within the results shown in Fig. 11 it can be seen that 5 operators are required to process the amount required within the time frame.

### 5.2.2. WIP

To demonstrate the ability of the simulation to update to changes in the real time data, a scenario is presented where a

customer wishes to know the time it will take to refurbish their products. This request is therefore based upon the current level of WIP in the factory. Information regarding the WIP within the remanufacturing business is sent from the traceability system to the simulation through a RESTful *GET\_WIP* service, highlighted in Table 3, with the values shown in Table 4. Simulation is then performed to determine the time taken for 3 operators to process the current WIP.

### 5.2.3. Production plan

The final example demonstrates the simulation of additional processes without changing the simulation constructs. A new process model developed within the Camunda modeller GUI to include additional processes for three products types (i.e. Mobile Phone, Laptop, Hard Drive), with a high-level depiction shown in Fig. 12. Historical data from the traceability system are analysed within the information services and imported into the simulation. Simulations constructs are not changed, instead the *Determine\_next\_activity* procedure assess the new process model and routes the products accordingly.

### 5.2.4. Discussion

The data-driven simulation presented within this paper has been implemented and demonstrated with the use of a case of involving WEEE refurbishment. It has been highlighted that the simulation constructs can automatically change based upon data inputs, provided by either the user or data captured within the shop floor system. This is beneficial as it doesn't require a user to make coding changes to the simulation manually when an aspect of the real-world changes.

However, there are a number of areas which haven't been considered within this research and require further work, which include:

- Logistics
- Control Logic
- Reducing initial set-up time

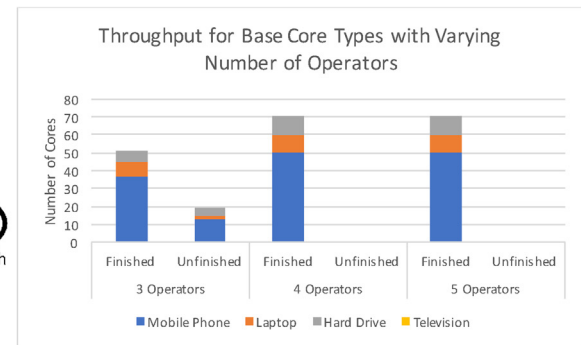
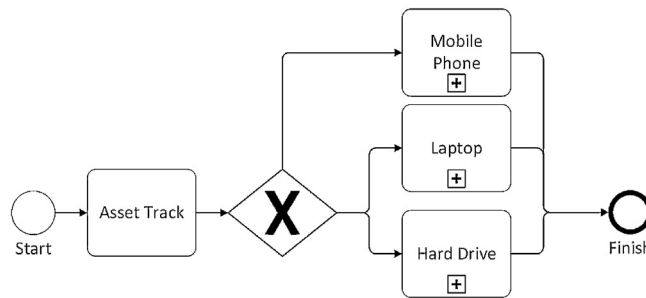


**Fig. 11.** Box and whisker plot comparing simulated throughput for different number of operators.

**Table 4**

Example of the WIP extracted from the traceability system.

Product Types	New order	Current WIP Process Stage					
		Asset Track	Visual Inspection	Functional Test	Data Erasure	Repair	Cleaning
Mobile Phones	33	20	5	13	15	3	7
Laptops	0	8	2	4	7	2	1
Hard Drives	0	14	3	5	8	1	2

**Fig. 12.** High level process model and simulation results for scenario for base cores.

Currently the system doesn't consider any logistical delays within the factory (i.e. the time and resources required to locate and move parts to locations within the factory to conduct an activity), instead it is assumed cores instantly move to the resources for processing. This was not reported within this work as it has been well researched by Wy et al. [7] and Wang et al. [10] however, further work will report on the integration of logistical aspects within the simulation. The control logic governing processing strategies has also not been the focus of this work. At present the simulation uses First In First Out (FIFO) logic to control the order in which cores are processed. Further research should be conducted into how material flow control strategies (i.e. KANBAN, CONWIP) can be integrated within the data-driven approach. Finally, whilst the maintenance of the data-driven simulation is improved by movement of the business logic away from the traditional approach of hard coding within, the initial set-up of the simulation can still require considerable time and resource, particularly to identify and retrieve data from digital systems to ensure data is in the correct format and is complete. Further work is in progress to investigate how technologies, such as ontologies and Artificial Intelligence (AI), can reduce the challenge of initial information discovery, retrieval and verification, whilst improved visualisation and user-interfaces could support any additional manual data entry.

## 6. Conclusion and future work

This paper outlines a data-driven simulation to predict material flow within remanufacturing operations. The novelty of this work is the removal of specific business logic from the hard-coded simulation to the input parameters, enabling it to be adapted and maintained simply to represent changes in the real world without the need to manually modify the simulation constructs. This is achieved through three main aspects:

- An *adaptive remanufacturing simulation* algorithm to model the complex material flow found within a remanufacturing process in a generic and reusable way.
- A *generic remanufacturing information model* to structure the data parameters of the simulation to represent remanufacturing features such as cores, factory resources and processes.

- An *Information Service Layer*, to collect, analyse and format sensor data from real-world factories and distributed sources for use in the simulation.

The simulation is implemented using a case study example of WEEE refurbishment to demonstrate its implementation and highlight how it can adapt to changes within the factory including the staff levels, WIP, and process plans. Data are input into the simulation from a combination of sources including real traceability RFID systems within the factory (historical data and WIP), manual user inputs (factory resources, simulation parameters) and third-party software interfaces (process plan).

This work aims to support concepts such as Industry 4.0 and the digital twin, by reducing the cost of maintaining and updating simulations to reflect changes within the real world. These can be used to support and optimise operational and tactical decisions and activities, such as production and process planning, in conjunction with other tools and algorithms.

A number of areas for future work have also been identified related to the inclusion of logistical aspects such as the time to locate and retrieve items for processing, extending the complexity of the control logic to include techniques such as KANBAN, and improving the ease of data discovery, retrieval and validation within the initial setup of the simulation. Finally, the application of the simulation could be advanced beyond remanufacturing to other manufacturing and after-market services such as maintenance.

## Acknowledgements

This work was supported by the Engineering and Physical Science Research Council (EPSRC) from funding for the project "AIIM" [grant reference EP/K014137/1].

## References

- [1] S.M. Jeon, G. Kim, A survey of simulation modeling techniques in production planning and control (PPC), *Prod. Plan. Control* 27 (5) (2016) 360–377.
- [2] A.A. Tako, S. Robinson, The application of discrete event simulation and system dynamics in the logistics and supply chain context, *Decis. Support Syst.* 52 (4) (2012) 802–815.

- [3] M.A. Ilgin, S.M. Gupta, Environmentally conscious manufacturing and product recovery (ECMPRO): a review of the state of the art, *J. Environ. Manage.* 91 (3) (2010) 563–591.
- [4] A. Negahban, J.S. Smith, Simulation for manufacturing system design and operation: Literature review and analysis, *J. Manuf. Syst.* 33 (2) (2014) 241–261.
- [5] S. Bergmann, S. Strassburger, Challenges for the automatic generation of simulation models for production systems, *Proc. 2010 Summer Simul. Multiconference*, No. Figure 1, (2010), pp. 545–549.
- [6] J.W. Fowler, O. Rose, Grand challenges in modeling and simulation of complex manufacturing systems, *Simulation* 80 (9) (2004) 469–476.
- [7] J. Wy, S. Jeong, B.I. Kim, J. Park, J. Shin, H. Yoon, S. Lee, A data-driven generic simulation model for logistics-embedded assembly manufacturing lines, *Comput. Ind. Eng.* 60 (1) (2011) 138–147.
- [8] D. Mourtzis, M. Doukas, D. Bernidaki, Simulation in manufacturing: review and challenges, *Proc. CIRP* 25 (C) (2014) 213–229.
- [9] P. Taylor, Y.T. Lee, F.H. Riddick, B. Johan, I. Johansson, Core Manufacturing Simulation Data – a manufacturing simulation integration standard: overview and case studies, *Int. J. Comput. Integr. Manuf.* 24 (8) (2011) 689–709.
- [10] J. Wang, Q. Chang, G. Xiao, N. Wang, S. Li, Data driven production modeling and simulation of complex automobile general assembly plant, *Comput. Ind.* 62 (7) (2011) 765–775.
- [11] P. Barlas, C. Heavey, Automation of input data to discrete event simulation for manufacturing: a review, *Int. J. Model. Simul. Sci. Comput.* 07 (01) (2016) 1630001.
- [12] M. Frantzen, A.H.C. Ng, P. Moore, A simulation-based scheduling system for real-time optimization and decision making support, *Robot. Comput. Integr. Manuf.* 27 (4) (2011) 696–705.
- [13] U. Ghani, R. Monfared, R. Harrison, Integration approach to virtual-driven discrete event simulation for manufacturing systems, *Int. J. Comput. Integr. Manuf.* 28 (8) (2014) 844–860.
- [14] S. Tavakoli, A. Mousavi, A. Komashie, A generic framework for real-time discrete event simulation (des) modelling, *Proceedings of the 2008 Winter Simulation Conference*, 2008, (1998), pp. 1931–1938.
- [15] S. Bagchi, C.H. Chen-Ritzo, S.T. Shikalgar, M. Toner, A full-factory simulator as a daily decision-support tool for 300mm wafer fabrication productivity, *Proc. - Winter Simul. Conf.*, (2008), pp. 2021–2029.
- [16] R. Azimi, S. Lee, S.M. AbouRizk, Intelligent look-ahead scheduling for structural steel fabrication projects, *Int. J. Adv. Software* 4 (1–2) (2011) 207–217.
- [17] V. Daniel, R. Guide, Scheduling with priority dispatching rules and drum-buffer-rope in a recoverable manufacturing system, *Int. J. Prod. Econ.* 53 (1) (1997) 101–116.
- [18] W.L. Ijomah, C.A. McMahon, G.P. Hammond, S.T. Newman, Development of design for remanufacturing guidelines to support sustainable manufacturing, *Robot. Comput. Integr. Manuf.* 23 (2007) 712–719.
- [19] A.M. King, S.C. Burgess, W. Ijomah, C.A. McMahon, Reducing waste: repair, recondition, remanufacture or recycle? *Sustain. Dev.* 267 (December 2005) (2006) 257–267.
- [20] R. Subramoniam, D. Huisin, R. Babu, Aftermarket remanufacturing strategic planning decision-making framework: theory & practice, *J. Clean. Prod.* 18 (16–17) (2010) 1575–1586.
- [21] M. Matsumoto, Y. Umeda, An analysis of remanufacturing practices in Japan, *J. Remanuf.* 1 (1) (2011) 2.
- [22] M. Lieder, A. Rashid, Towards circular economy implementation: a comprehensive review in context of manufacturing industry, *J. Clean. Prod.* 115 (2016) 36–51.
- [23] V.D.R. Guide, Production planning and control for remanufacturing: Industry practice and research needs, *J. Oper. Manag.* 18 (4) (2000) 467–483.
- [24] B. Esmailian, S. Behdad, B. Wang, The evolution and future of manufacturing: A review, *J. Manuf. Syst.* 39 (2016) 79–100.
- [25] P. Goodall, E. Rosamond, J. Harding, A review of the state of the art in tools and techniques used to evaluate remanufacturing feasibility, *J. Clean. Prod.* 81 (2014) 1–15.
- [26] M.R. Galbreth, J.D. Blackburn, Optimal acquisition quantities in remanufacturing with condition uncertainty, *Prod. Oper. Manag.* 19 (1) (2010) 61–69.
- [27] X. Zhang, H. Zhang, Z. Jiang, Y. Wang, An integrated model for remanufacturing process route decision, *Int. J. Comput. Integr. Manuf.* 28 (5) (2014) 1–9.
- [28] E. Sundin, B. Bras, Making functional sales environmentally and economically beneficial through product remanufacturing, *J. Clean. Prod.* 13 (9) (2005) 913–925.
- [29] H.-B. Jun, M. Cusin, D. Kiritsis, P. Xirouchakis, A multi-objective evolutionary algorithm for EOL product recovery optimization: turbocharger case study, *Int. J. Prod. Res.* 45 (August 2015) (2007) 4573–4594.
- [30] C. Li, Y. Tang, C. Li, L. Li, A modeling approach to analyze variability of remanufacturing process routing, *IEEE Trans. Autom. Sci. Eng.* 10 (1) (2013) 86–98.
- [31] H.C. Zhang, J. Li, P. Shrivastava, A. Whitley, M.E. Merchant, A web-based system for reverse manufacturing and product environmental impact assessment considering end-of-life dispositions, *CIRP Ann. Manuf. Technol.* 53 (1) (2004) 5–8.
- [32] K. Xing, L. Luong, Modelling and evaluation of product fitness for service life extension, *J. Eng. Des.* 20 (3) (2009) 243–263.
- [33] S. Behdad, A.S. Williams, D. Thurston, End-of-life decision making with uncertain product return quantity, *J. Mech. Des.* 134 (10) (2012) 100902.
- [34] Department for Business Innovation & Skills, PAS 141:2011, Reuse of used and waste electrical and electronic equipment (UEEE and WEEE), Process Management. Specification, British Standards Institution, 2011.
- [35] Y. Du, H. Cao, F. Liu, C. Li, X. Chen, An integrated method for evaluating the remanufacturability of used machine tool, *J. Clean. Prod.* 20 (1) (2012) 82–91.
- [36] P.B. Luh, D. Yu, S. Soorapanth, A.I. Khibnik, R. Rajamani, A lagrangian relaxation based approach to schedule asset overhaul and repair services, *IEEE Trans. Autom. Sci. Eng.* 2 (2) (2005) 145–156.
- [37] C. Franke, B. Basdere, M. Ciupek, S. Seliger, Remanufacturing of mobile phones-capacity, program and facility adaptation planning, *Omega* 34 (6) (2006) 562–570.
- [38] J. Li, M. González, Y. Zhu, A hybrid simulation optimization method for production planning of dedicated remanufacturing, *Int. J. Prod. Econ.* 117 (2) (2009) 286–301.
- [39] L. Gaspari, L. Colucci, S. Butzer, M. Colledani, R. Steinhilper, Modularization in material flow simulation for managing production releases in remanufacturing, *J. Remanuf.* 7 (2–3) (2017) 139–157.
- [40] P.S. Muldamala, J.S. Smith, J.F. Valenzuela, Designing reusable simulation modules for electronics manufacturing systems, *Simul. Conf. 2003. Proc. 2003* (2003) pp. iii–xxiii.
- [41] F.M. Martínez, L.M.A. Bedia, Modular simulation tool for modelling JIT manufacturing, *Int. J. Prod. Res.* 40 (7) (2002) 1529–1547.
- [42] S. Robinson, Conceptual modelling for simulation Part II: a framework for conceptual modelling, *J. Oper. Res. Soc.* 59 (3) (2008) 291–304.
- [43] Y. Huang, M.D. Seck, A. Verbraeck, From data to simulation models: component-based model generation with a data-driven approach, *Proceedings of the 2011 Winter Simulation Conference*, (2011), pp. 3724–3734 no. 2010.
- [44] B.R. Ferrer, B. Ahmad, A. Lobov, D.A. Vera, J.L.M. Lastra, R. Harrison, An approach for knowledge-driven product, process and resource mappings for assembly automation, *IEEE Int. Conf. Autom. Sci. Eng.* 2015 (October) (2015) 1104–1109.
- [45] A. Matsokis, D. Kiritsis, An ontology-based approach for product lifecycle management, *Comput. Ind.* 61 (8) (2010) 787–797.
- [46] J. Um, J.-S. Yoon, S.-H. Suh, An architecture design with data model for product recovery management systems, *Resour. Conserv. Recycl.* 52 (10) (2008) 1175–1184.
- [47] O.M. Group, Object Management Group Business Process Model and Notation [Online]. Available: (2016) . <http://www.bpmn.org/>.
- [48] R.G. Sharpe, P.A. Goodall, A.D. Neal, P.P. Conway, A.A. West, Cyber-Physical Systems in the re-use, refurbishment and recycling of Used Electrical and Electronic Equipment, *J. Clean. Prod.* (2017) In Press.
- [49] Holusko Software, “Delsi 2.0.” [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=HolushkoSoftware.Delsisimulationssystem#overview> (Accessed 30 June, 2017).
- [50] Camunda, Camunda BPM platform - Overview, 2017. [Online]. Available: <https://camunda.com> (Accessed 07 August, 2017).