Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-023-02257-3

Automatic model generation for material flow simulations of Third-Party Logistics

Lennart M. Steinbacher[1] · Tim Düe[1,2] · Marius Veigt[1] · Michael Freitag[1,2]

Received: 30 March 2023 / Accepted: 25 October 2023

Abstract

The use of Third-Party Logistics (TPL) is a common practice among manufacturing companies seeking to increase profitability. However, the tender process in selecting a TPL service provider can be challenging, requiring significant effort from both the tendering company and the service provider. The latter must meticulously plan processes and calculate pricing positions while running the risk of losing the bid. This risk impedes verifying logistical feasibility and comparing different logistic concepts extensively, such as layouts, which are often work-intensive. With the ongoing progress of research toward automatic simulation model generation for material flow, it is left to answer whether such approaches can improve the planning processes of TPL service providers by using planning data to generate simulation models. Therefore, this work presents a system with an underlying ontology to generate material flow simulations by developing a model transformation methodology. The system's functions are tested to determine whether they can support the planning process using defined case studies that cover everyday planning decisions. As a result, the system is capable of verifying the performance of planned logistic systems with minimal manual modelling efforts. This encompasses the evaluation of alternative logistical concepts for configuring the planned systems.

Keywords Third-Party Logistics · Material flow · Automatic simulation model generation · Tendering

Introduction

The Third-Party Logistics (TPL) industry has experienced steady growth in recent decades. Despite recent volatility, outsourcing logistics operations to TPL providers remains a common strategy for improving profitability for Original Equipment Manufacturers (OEMs). As such, a considerable amount of literature has been published on the subject, with many definitions provided, including the one put forward by Premkumar et al. (2021), which describes it as mutually beneficial relationships between supply chain interfaces and TPL providers, with logistics services offered in shorter or longer-term partnerships, ranging from basic to highly customised services with a focus on efficiency and effectiveness. Customised services, in particular, are critical for industrial customers (Large 2011).

In the face of increasing competition due to digital disruption, TPL providers must reduce costs to remain competitive (Hofmann and Osterwalder 2017), and one approach to do so is through the tender process as it has been identified as a crucial yet problematic element of TPL business processes (Peters et al. 1998). CEOs of TPL providers have expressed concerns regarding the effort required in terms of time and qualified personnel to respond to tender invitations. To better understand the tender process, Straube et al. (2011) analyzed the tender management process and identified two core processes: designing TPL services concepts and calculating costs. These processes require highly qualified personnel and

✉ Lennart M. Steinbacher
stb@biba.uni-bremen.de

Tim Düe
due@uni-bremen.de

Marius Veigt
vei@biba.uni-bremen.com

Michael Freitag
fre@biba.uni-bremen.com

[1] BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Hochschulring 20, 28359 Bremen, Germany

[2] University of Bremen, Faculty of Production Engineering, Badgasteiner Straße 1, 28359 Bremen, Germany

Published online: 07 December 2023

contribute to the knowledge-intensiveness of the overall tender process (Rajesh et al. 2011; Ristić and Davidović 2019).

Currently, standard office tools, specifically spreadsheet tools, are commonly used in these processes, and companies have started to enrich them with database applications (Ristić and Davidović 2019; Straube et al. 2011). Standardisation through modularisation has also been applied to the case of contract logistics with the development of software tools (Spiegel et al. 2014), while supervised machine learning algorithms have been developed to support planning personnel in the design processes of TPL service concepts (Veigt et al. 2022). Their approach suggests Method-Time Measurement (MTM) steps for specific logistical tasks based on meta-information of the tender project and prior defined process steps. Due to the temporal shift between designing a tender and realising a business, planning personnel cannot receive direct feedback on whether the resource allocation and layouts have met the customer's requirements. Additionally, strict time limits and resource scarcity prevent planning personnel from exploring various alternative layouts. This paper contributes to a framework and a proof of concept for improving logistic service design by automatically deriving material flow simulations from spreadsheet planning information using model transformation.

This paper is structured as follows: Section "Overview" presents the current state of the art in the fields of automatic simulation model generation (ASMG). In Section "Simulation framework", the proposed ontology as well as the methodology of model transformation is described in the form of an algorithmic procedure. Next, in Section "Demonstration" the suitability of the ontology for intralogistics applications in TPL is tested in conjunction with the model transformation procedure. Here, the illustrative case studies used for evaluation are elaborated. The results of those case studies are described and discussed in Section "Evaluation". Section "Conclusion and Outlook" gives a conclusion and outlook of the presented research.

This paper is an extension of the article from Steinbacher et al. (2022). This article presented a framework for modelling production systems to train reinforcement learning algorithms for control tasks in mere production environments. The underlying ontology was adjusted to the TPL applications. Further, a new simulation framework was developed, including the necessary functionalities to model and simulate TPL systems. Consequently, the main contribution in extending to the prior research, is the generalization for TPL applications in Section "Ontology", a generalized system design for model transformation in Section "System design", and finally the model transformation procedure (Section "Modelling") from MTM steps to aggregated yet accurate processes.

Overview

Discrete event simulation (DES) is a well-established method in the manufacturing and logistics industry for modelling work and material flows, also known as material flow simulations (MFS) (Barlas and Heavey 2016). MFS models are used to evaluate system performance under dynamic influences, with applications in various industries, including hospital material flows (Fragapane et al. 2019) and port logistics (Hoff-Hoffmeyer-Zlotnik et al. 2017). MFS is frequently used in the layout planning of industrial facilities (Centobelli et al. 2016). For instance, Centobelli et al. (2016) used MFS to optimise the layout of a digital factory to improve its logistical performance. Similarly, MFS has been used to evaluate reconfiguration scenarios for production systems (Hoellthaler et al. 2019). MFS has been identified as a valuable tool for software evaluation criteria for rapid layout evaluation planning (Shariatzadeh et al. 2012).

Besides layout planning, MFS is a common method for validating approaches and policies in production control. For example, Eberlein and Freitag (2022) tested a pull principle for material supply in the aerospace industry using MFS. Steinbacher et al. (2022) developed a simulation framework using MFS to train reinforcement learning algorithms to schedule production systems dynamically. Gaspari et al. (2017) standardised the overall approach by modularising MFS to evaluate control policies in remanufacturing. Another area where MFS is applied concerns digital twins for production systems. For instance, Uhlemann et al. (2017) used MFS to validate scenarios for optimising digital twins. Furthermore, Müller et al. (2021) used MFS inside digital twins to control subsystems like autonomous guided vehicles. Another simulation application in digital twins is a virtual test environment, for example, to configure interfaces between physical systems and the virtual twins (Ait Alla et al. 2020).

All these applications of MFS rely on valid simulation models, which require considerable effort to create. Thus, ASMG is a sought-after solution and a recurring research topic Reinhardt et al. (2019). Reinhardt et al. (2019) surveyed works with this research goal and established a classification scheme based on the data source, data variability over time, information retrieval, and standardisation. Some examplary works in this area include Lütjen and Rippel (2015), who built a framework for modelling and simulating complex production systems from user interfaces and a knowledge database, and Bergmann (2014), who retrieved necessary information from enterprise databases and knowledge databases with the help of machine learning. In this regard, Bergmann and Straßburger (2015) describe lessons learned with so-called core manufacturing simulation data (CMSD).

In addition to the survey by Reinhardt et al. (2019), several other works have explored the automatic generation of material flow simulations. Milde and Reinhart (2019) developed a data-based approach that utilises tracking and error data to identify the material flow, parameterise it, and identify control policies while generating an exchange format and a model generator for the simulation software Tecnomatix Plant Simulation. Lugaresi and Matta (2020) present a similar approach that includes an additional validation step at the end. Using semantic knowledge, Jurasky et al. (2021) proposed an approach that applies mapping rules for a use case ontology and simulation ontology and then generates a model with the web ontology labels standard. This model can then be parsed to create a model for the simulation software AnyLogic model. Vernickel et al. (2020) developed an approach that includes manual model generation with machine-learning-based support to synchronise and parameterise the simulation model.

However, to our knowledge, no published work has yet been able to generate simulation models from planning data provided in the early stages of TPL tender process. Previous approaches have relied on data from operations to generate or parameterise the model. Consequently, we formulate the following research question: Is an ASMG capable of improving the planning process of TPL?

Simulation framework

Ontology

As previously mentioned, CMSD is an established information model that the Simulation Interoperability Standards Organization (SISO) has standardised with the aim of promoting the interoperability of simulation models with other enterprise information systems. It comprises a collection of Unified Markup Language (UML) classes. It includes sub-classes such as *layout*, *part information*, *resource information*, *production operations*, *production planning*, and *support* (Group 2010a). The Extensible Markup Language (XML) representation of CMSD allows for integrating related data into other systems (Group 2010b). While CMSD has been widely adopted, it has some limitations that require attention in future editions, such as the representation of resource capacities (Bergmann and Straßburger 2015). Additionally, the representation of jobs and schedules in CMSD diverges from the requirements of TPL applications, which focus on factory supply rather than job shop settings. As a result, a more simplistic ontology tailored to the specific needs of TPL applications has been developed using the UML standard. This proposed ontology is divided into four packages, each depicted in Figs. 1 to 4 along with their respective classes.

The first package, depicted in Fig. 1, includes the abstract superclass `Resource` and its sub-classes: `StorageRes`, which defines storage resources with the time needed to release a stored object; `ImmobileRes`, which defines resources that cannot be moved; and `TransportRes`, which is implemented for any resource that can transport objects and defines the loading time of an object. `TransportRes` further inherits the class `ImmobileTransportRes`, which represents static transportation resources like conveyor belts. They are associated with at least two instances of the class `AnchorPoint` to define their location. The subclass `MobileRes` defines mobile resources, such as moving equipment, along with their respective speed. A new subclass, `MobileTransportRes`, is extended from the combination of `TransportRes` and `MobileRes`. This subclass represents movable equipment and is also a transport resource, for example forklifts. Both `MobileTransportRes` and `ImmobileTransportRes` are associated with the class `TransportType`, which declares the type of goods that can be transported. Instances of `TransportType` are then associated with instances of `PhysicalObject` to define whether a transport resource can transport an object. The class `Fleet` is a composition of the classes `MobileRes` and `MobileTransportRes`. The class `Worker`, which represents a kind of resource, is associated with the class `WorkerType`, which, in turn, is associated with the class `Parking`. This applies to the class `Fleet` as well. The class `Parking` defines the location where unoccupied resources wait for their next task by associating it with an instance of the class `AnchorPoint`.

In Fig. 2, the second package of the ontology is presented, which deals with organisational aspects. The central class of this package is `Shift`, which defines the active period of the logistical system by specifying the daytime and workdays on which the shift is active. This class is associated with the `WorkDays` enumeration. The `Shift` class is also associated with instances of the class `Dispatching`, which defines the spawning of objects in the system, their location, and their statistical distribution in time. This enables the simulation of different system load conditions. The `Dispatching` class includes an enumeration called `SpawnTypes` which currently has three options: `set_times`, `frequency` and `distribution`. The `set_times` option specifies dispatches at specific points in time, such as when a truck of to-be-processed products arrives at a particular time. The `frequency` option sets a defined frequency, for example, when a production line outputs products at a specific rate and requires further logistical services.

In Fig. 3, we define the operational aspects of the simulation's elements. This package's centre is the `Process` class, which encompasses all necessary associations and attributes. This class defines the inputs and outputs of the process, as well as the resources it requires. The `ImmobileTransportRes` resource is a regular resource outside an usual station. In addition to inputs, outputs and

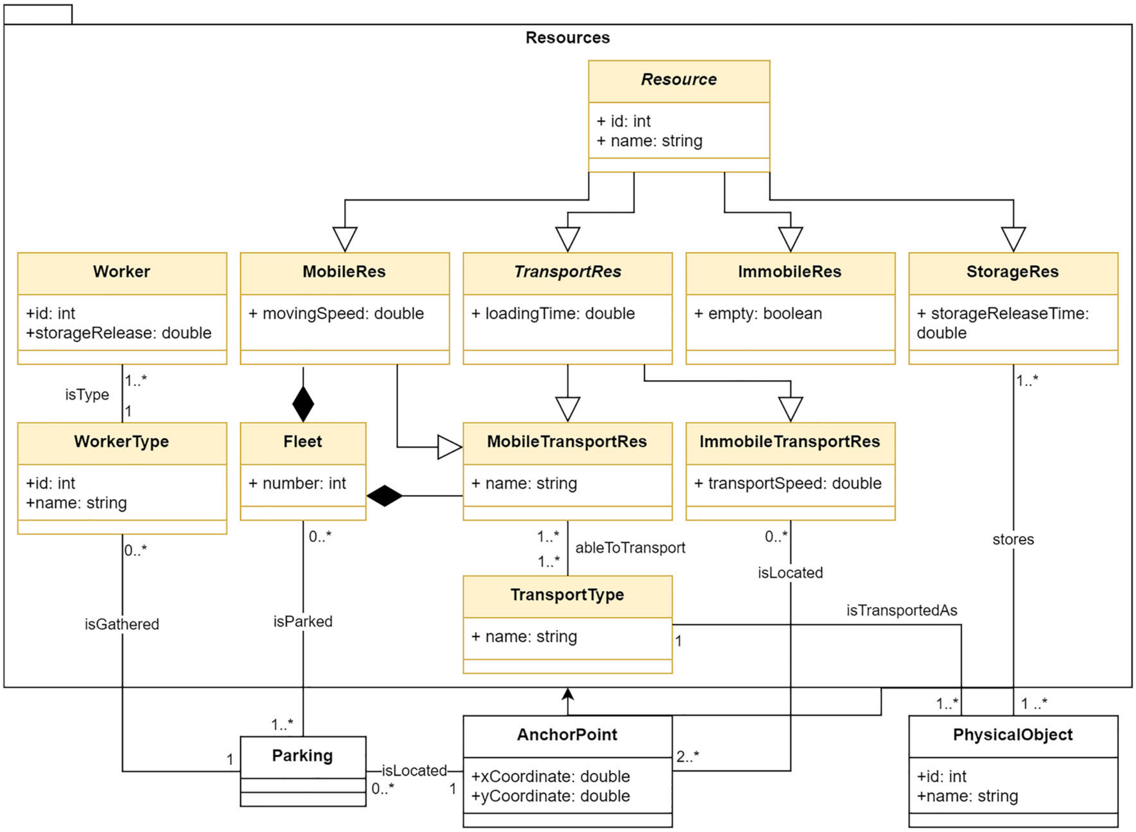


Fig. 1 Resources package

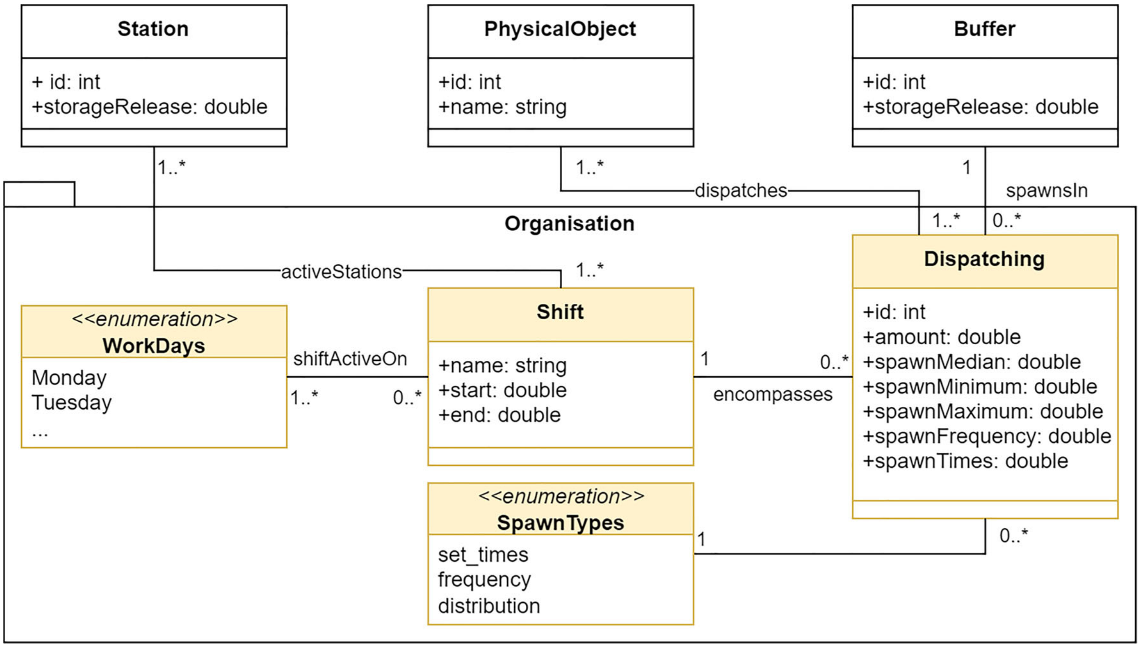


Fig. 2 Organisation package

resources, it is also crucial to establish process chains. This is accomplished through the ProcessLink class, which establishes associations between instances of ProcessLink and processes, defining an origin and a process destination. Each link also contains the attribute probability, allowing for diverging process graphs. To start a process chain, a physical object is associated with an instance of a start process as initialProcess, defined by the process chain from then on. The duration of the process is defined through attributes of the class and the association to an enumeration instance of DistributionTypes. Finally, the SuperProcess class describes types of processes and technical processes that are included due to specific markers in process chains, such as start and end processes.

In Fig. 4, the package describing the layout of the ontology is presented. This package comprises of several classes, including the AnchorPoint class, which is responsible for defining the location of elements in a two-dimensional Cartesian coordinate system. The central element of this package is the Station class, which is associated with at least one input instance and one output instance of the Buffer class. Additionally, the Station class is associated with the StationType class, which acts as a blueprint for creating multiple identical stations without defining each one individually. The StationType class includes information about the resources and processes associated with each station type.

In addition to the station types, the Buffer class is associated with an instance of the AnchorPoint class to define its location. The Buffer class also includes information about its capacity to hold specific objects. Finally, the Path class defines the routes any movable equipment or object can take within the system.

System design

The system in Fig. 5 is designed to enable the integration of the component User into the TPL concept design process. To calculate the cost of a project, the process planning involves creating and evaluating processes and resources. The planning personnel, represented by the component User, can generate the calculation sheets independently or with the aid of the Assistance Tool proposed by Veigt et al. (2021). The resulting calculation of a TPL project is an text file called Project Calculation, which contains information about resources, organisation, operation, and layout. This information can be edited by the Analysis component in the Model Transformation subsystem, implemented using the Eclipse Modelling Framework (EMF). The Analysis component analyses the MTM steps in the Project Calculation, including information about resources, personnel, and transport processes, to generate a model based on the presented ontology. However, due to inconsistencies among planning personnel, the proposed model needs to be modified by the users to ensure consistency and accuracy.

Once the proposed analysis phase is finished, the Import component reads the proposed and edited model to generate an EMF model. The Modelling component then uses this model to implement a user interface that allows the users to make adjustments, such as alternative layouts or process graphs. The Export component accesses the EMF model generated by the Modelling component. It creates a JSON file that contains the necessary EMF model-related objects and attributes required for simulation.

Outside the Modelling subsystem, a simulation application has implemented an ASMG approach for TPL projects. This application uses the commercial simulation software AnyLogic 8.7 and provides a graphical interface to verify and edit the model in case of deviations. While running the model, a continuous stream of information is buffered and written to a text file, which the Dashboard component uses to extract knowledge from different alternative concepts. For example, the Dashboard component helps the planners determine if specific dynamic effects compromise the offer to the customer of the TPL project. Overall, this system enables efficient and accurate TPL project planning and management.

Modelling

The Modelling sub-system is a complex IT system comprising four components. Of these four components, the Analysis component is particularly relevant in answering the research rather than just technical ones. To provide a detailed explanation of the Analysis component, an overview of fictitious process models during different analysis phases is presented in Fig. 6. This figure shows the different levels of the process, which are structured from top to bottom. The highest level, called the main process, typically bundles processes with some form of spatial proximity, such as goods receipt. The level below, called the process, bundles sub-processes specific to a particular object type for which a price position needs to be calculated. For example, when a particular good is delivered to the logistic site, processes specific to this good are bundled into a process. The third level, named sub-process, bundles the atomic process steps defined by an MTM code. The first of two analysis phases, as shown in the lower part of Fig. 6, transforms the process steps into so-called aggregated processes (AP), whose procedure is later described in more detail. To do so, information from the artifact Configuration File is retrieved to bundle process steps that can be simulated. The second phase creates a Petri-net model using this information and the AP structure from the phase.

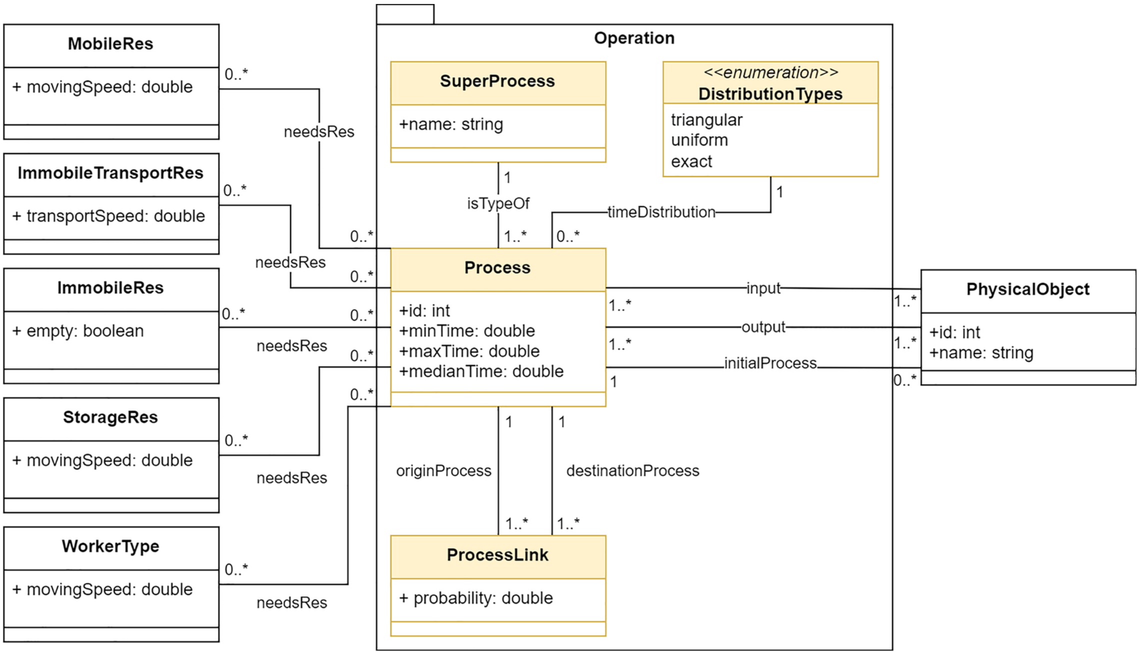


Fig. 3 Operation package

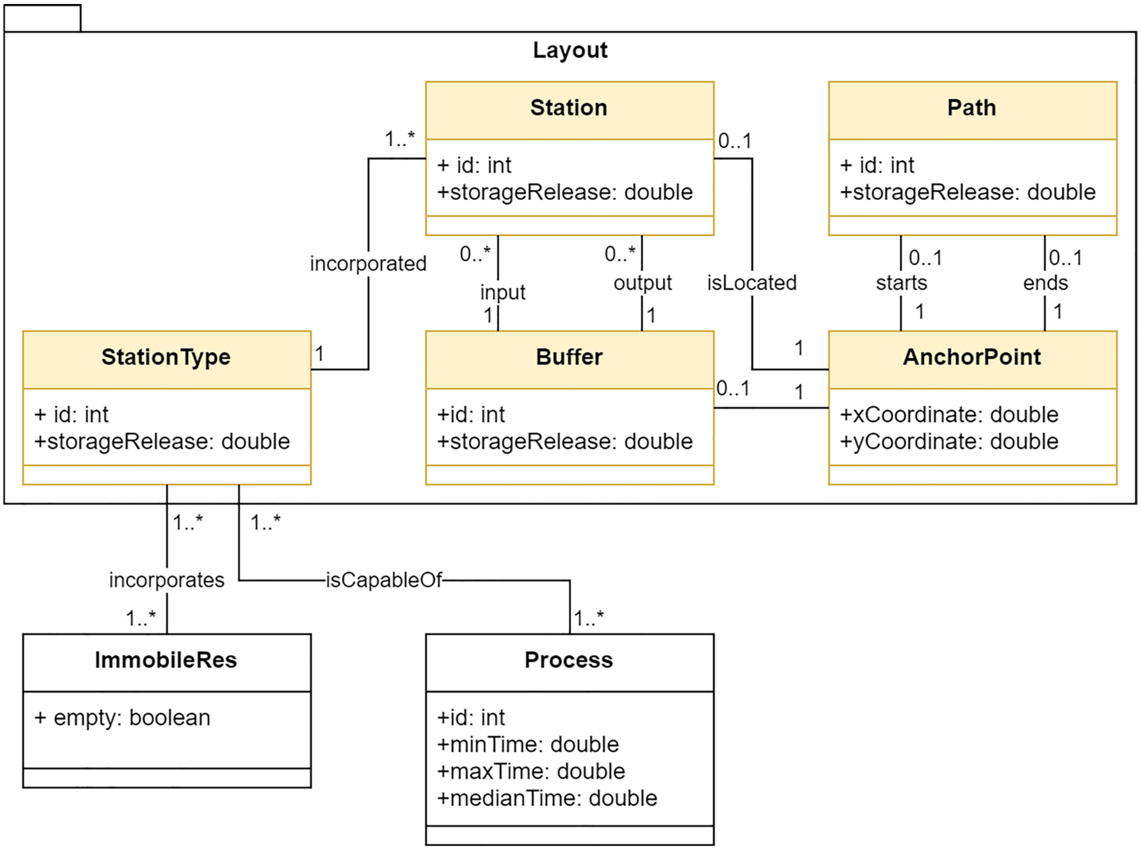


Fig. 4 Layout package

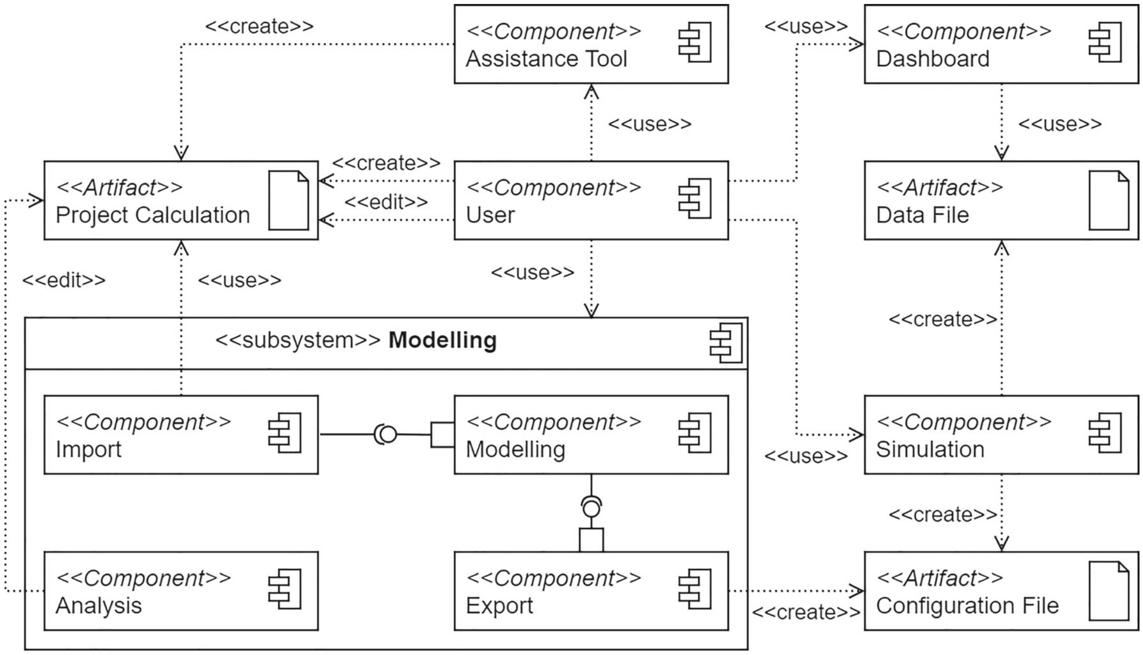
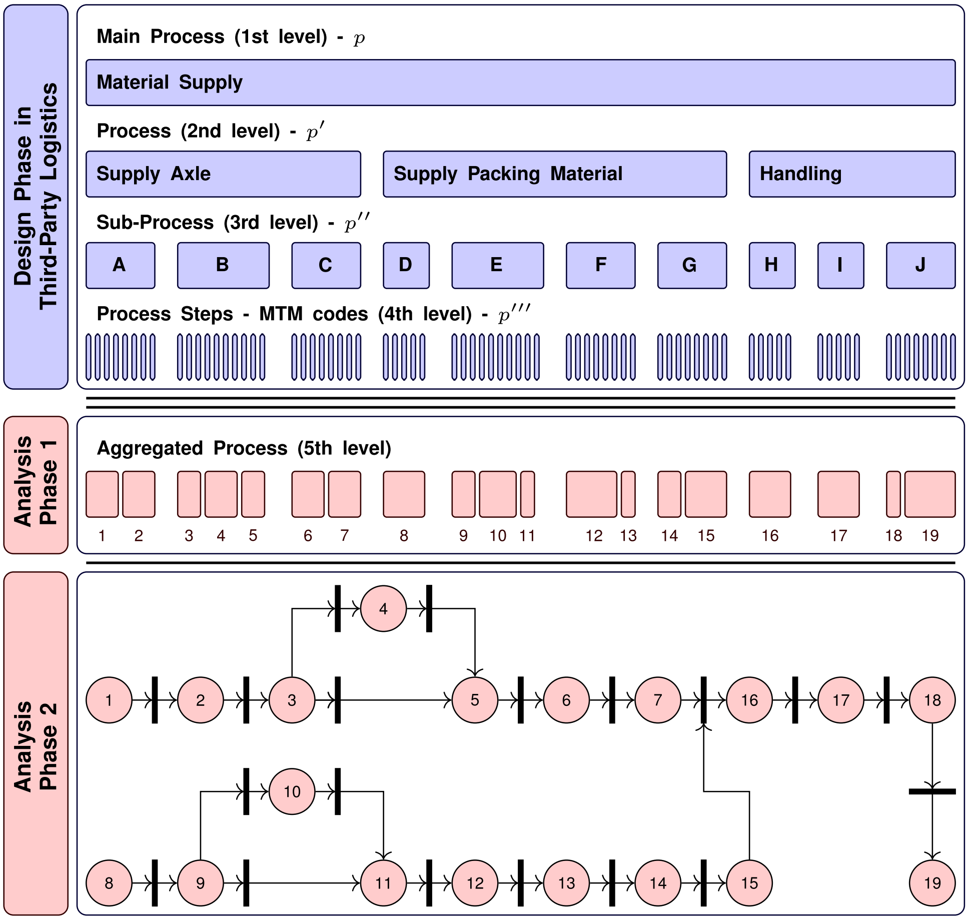


Fig. 5 System components and their interfaces

Fig. 6 Process models during the TPL planning process as well as different phases of the Analysis


Design Phase in Third-Party Logistics
Main Process (1st level) - p
Material Supply
Process (2nd level) - p'
Supply Axle
Supply Packing Material
Handling
Sub-Process (3rd level) - p''
A
B
C
D
E
F
G
H
I
J
Process Steps - MTM codes (4th level) - p'''
Analysis Phase 1
Aggregated Process (5th level)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Analysis Phase 2
4
1
2
3
5
6
7
16
17
18
10
8
9
11
12
13
14
15
19

The presented process hierarchy is agnostic to the TPL providers' specific design and calculation processes or process hierarchy as the lowest level of the TPL process structure is defined by `process steps` with their specific MTM codes, which could also be exchanged for other work design methods. Thus, the approach can be transferred to different TPL service providers. Although, it may appear that further information linked to these process steps can differ between different TPL service providers. Nevertheless, the shown example in Table 1 includes columns with information typical in the TPL concept design process. The column `Unit` refers to the object to which the process step relates. The `probability` column includes information about process steps that only occur at a specific rate or chance. The columns `factor` and `divisor` exist due to unit-specific pricing positions. The values in these columns are used to calculate the time expenditures according to their correct extent correctly. The tendering company compares different tenders by the unit pricing positions in a particular process. However, due to process steps that occur multiple times per unit or only at a specific batch size, these process steps need to be multiplied to attribute the correct cost. An example is palletisation, where a product is stacked on a pallet four times. When every package is commissioned on the pallet, an additional process step could be performed once per pallet. Without a divisor, this last pallet-wise process would be calculated for every package, leading to incorrectly calculated processes. Finally, the `resource` and `worker type` columns define the resources or worker types needed, while the `aggregated process` (AP) column contains the results of the first of two analysis phases.

In the first phase, every `sub-process` is divided into APs by identifying uniform subparts in terms of `probability`, `resource` and `worker type`. This uniformity is a prerequisite so that no resource or worker is longer occupied in a simulated process than needed, this functionality is defined as function **R** in the Algorithm 1. This holds especially true for deviating probabilities of process steps as those indicate the diverging processes. Another part of the first analysis phase is the identification of transports. Here different sets of MTM codes are used to indicate transportation processes.

There are common transport codes, like `SFKSF`, or placeholder codes, like `PT`. But more is needed to identify them and detect the proper context. For example, `PT` and `SZAGF` (surcharge for using a long fork in forklift) are ambiguous codes that also could be used in handling processes. Therefore a more sophisticated approach is needed. For this, the detection method uses the factored MTM codes representing transportation. Codes like `SFISF` are paired with factors to indicate the distance they need to be transported. So whenever a cluster of transportation codes is found in the spreadsheet, the method checks if a factor indicates a considerable transportation distance. The magnitude of this factor differentiates from project to project. The lower boundary for this factor indicates whether this transportation cluster is a transport or a station's handling process. So setting this factor to the shortest distance between any stations in the system is a good approximation. After these rules, a set of aggregated processes from the process level exist, as presented in the column `aggregation process` in Table 1.

The second phase of the analysis performs the model transformation from APs to a Petri-net process model. The main issue when processing APs, is finding the correct inputs, outputs and overall process graph. These issues are based on the fact that concept design and calculation in TPL usually do not follow standard process notation, like Business Process Management Notation, which would allow classic model transformation techniques. Thus, custom tests were developed to transform the MTM codes' properties into a Petri-net process model. The starting point is the first AP. Its inputs are derived from the inputs of the super-ordinate process. From then on, the algorithms loop through the APs. In each AP, two separate tests with conditional logic are applied. Figure 7 depicts both decision trees. The first decision tree (a) tests in the first condition if the input objects of the `process` are also the output object of that `process` and if they are collectively exhaustive, meaning there are no other objects as outputs. If this is the case, no more testing is done, and every AP gets the same input and output objects. The second condition tests if a unit is inside the current AP that needs to be simulated in the material flow. This classification is called material flow object (MFO). This is done by testing if a particular object dominates any APs in this overall `process`. Aside from the classification MFO, objects can also be classified as auxiliary objects (AO). AOs are objects, or as is described in Table 1, `units`, which are the reference object for specific MTM codes but are not simulated as a physical element in the simulation, e.g. cargo lists. The allocation of resources and workers is attributed to the MFO. So if the MFO exists, it will be used as the output of the AP. Accordingly, the specific AP acts as a transformation process which uses the given inputs to create the MFO as an output. When those inputs for transformation appear later in other APs, they are now categorised as AOs and are not simulated accordingly, as the AP used these MFOs to create other MFOs. But if there is no MFO, the third conditional logic is applied. With the function `InputsAppearance`, which tests whether any later APs consist mainly of input objects. If this is the case, these objects seem to be an MFO and appear as input later. With this reasoning, the given inputs of an AP are also regarded as an output. Suppose this is not the case with the fourth conditional logic test if any MFOs are in the following APs. In this case, a new dummy `process step` with this MFO is artificially added to recognise it as an output of this AP. But if even this last condition is not fulfilled, the

Table 1 Illustrative spreadsheet data from typical TPL processes after applying the first analysis phase with aggregated processes as a result

Process	Subprocess	Process Step	MTM	Unit	Prob.	Factor	Divisor	Resource	Worker	Agg. Process
Reception pallet	Reception package	walk_to_truck	KA	Truck					Booker	
		talk_to_driver	PT	Truck					Booker	
		check_truck	PT	Truck					Booker	agg_booking_1
	Material reception	take_list	IAHO3	Truck				Scanner	Clerk	
		scan_list	IDES	Truck				Scanner	Clerk	agg_scan_1
		register_truck	IDES	Truck					Clerk	agg_register
	Truck unloading	check_pallet	PT	Pallet			2	Forklift	Clerk	agg_unload
		pick_up_pallet	SABAFM	Pallet		3	2	Forklift	Clerk	
		drive_to_buffer	SFISF	Pallet		45	2	Forklift	Clerk	(transport_1)
		document_shortage	IAKW	Pallet	10%		2		Clerk	
		document_damage	IAKW	Pallet	10%		2		Clerk	agg_quality
Reception package	Entrance	register_courier	PT	Package					Receipt	
		talk_to_courier	PT	Package					Receipt	
		check_list	IAHO3	Package					Receipt	agg_entrance_1
		document_shortage	IAKW	Package	5%				Receipt	
		document_damage	IAKW	Package	5%				Receipt	agg_entrance_2
		pick_up_package	HUDT	Package					Receipt	
		arrange_pallet	HUDT	Pallet					Receipt	
		secure_pallet	HUDT	Pallet					Receipt	agg_handling_1
	Material handling	load_pallet	HUDT	Pallet					Clerk	
		take_to_carrier	KA	Pallet					Clerk	agg_handling_2
		transport_to_buffer	SFISF	Pallet				Cart	Clerk	(transport_2)
Storage		pallet_to_pallet	HUDT	Pallet					Clerk	agg_handling_3
	Material handling	drive_to_storage	SFISF	Pallet		60	2	Forklift	Clerk	
		turning_manoevre	SFKLF	Pallet		5	2	Forklift	Clerk	(transport_3)
	Booking	take_scanner	IAHO3	Pallet				Scanner	Clerk	
		scan_pallet	IDES	Pallet				Scanner	Clerk	
		scan_storage	IDES	Pallet			2	Scanner	Clerk	agg_booking_2

AP uses the object from the overall `process` as MFO. This constitutes the first decision tree, which defines the inputs and outputs of APs.

The decision tree (b) of Fig. 7 makes generating the process model as a Petri-net possible. `Processes` are often spatially concentrated at a particular station or area in the system and related to a specific object. Thus diverging process graphs are usually limited to additional process notes that fuse back into the linear process. In the first conditional logic, the existence of the probabilities is checked. If it's not the case, then it is assumed that a linear process graph with no deviating processes exists. Otherwise, the second conditional logic tests whether there are APs with complimentary probabilities. For instance, if there is an AP with a 20% probability, it searches for another AP, constituting an 80% probability. If no such APs have complimentary probabilities, an additional transition is added to the linear, directed graph surpassing the AP with the given probability. On the contrary, if APs have complimentary probabilities, whether those APs are transport processes is tested. If so, the same measures are taken as in the initial condition, and a transition is created that surpasses the AP with probabilities. New branches of transitions with processes with parallel structures are only created if those complementary processes are not transportation processes, as transportation processes are not directly modelled but are simulated if successive processes need to transport from one to another station. Applying the second decision tree completes the second analysis phase, generating a Petri-net. A pseudo-code overview of the `analysis` is given with the Algorithm 1.

Automatic simulation model generation

There are mainly two ways to generate simulation models automatically in AnyLogic. First, the XML elements of the `.alp` file, which stores AnyLogic simulation models, is edited according to the applications' needs. The second option is to build simulation models by explicitly initialising the network and level class and populating it with agents. The latter is used as it offers clarity towards large models. The `export` component of the sub-system `modelling` generates the artefact `Configuration File` with the JSON format. This necessary information to create the simulation is initialised to an object in the Java environment of AnyLogic. Afterwards, a set of functions is called, generating the simulation elements based on the information in the `Configuration File`. This includes creating a general simulation layer and adding parts like processes, shifts, and navigation paths. It also includes adding agents that each has a state-based control. These agents include transporters, products, stations, buffers and spawners, which create products and add them to the simulation. Lastly, some setup functions run to set up the resources, shifts, and buffer-to-station rela-

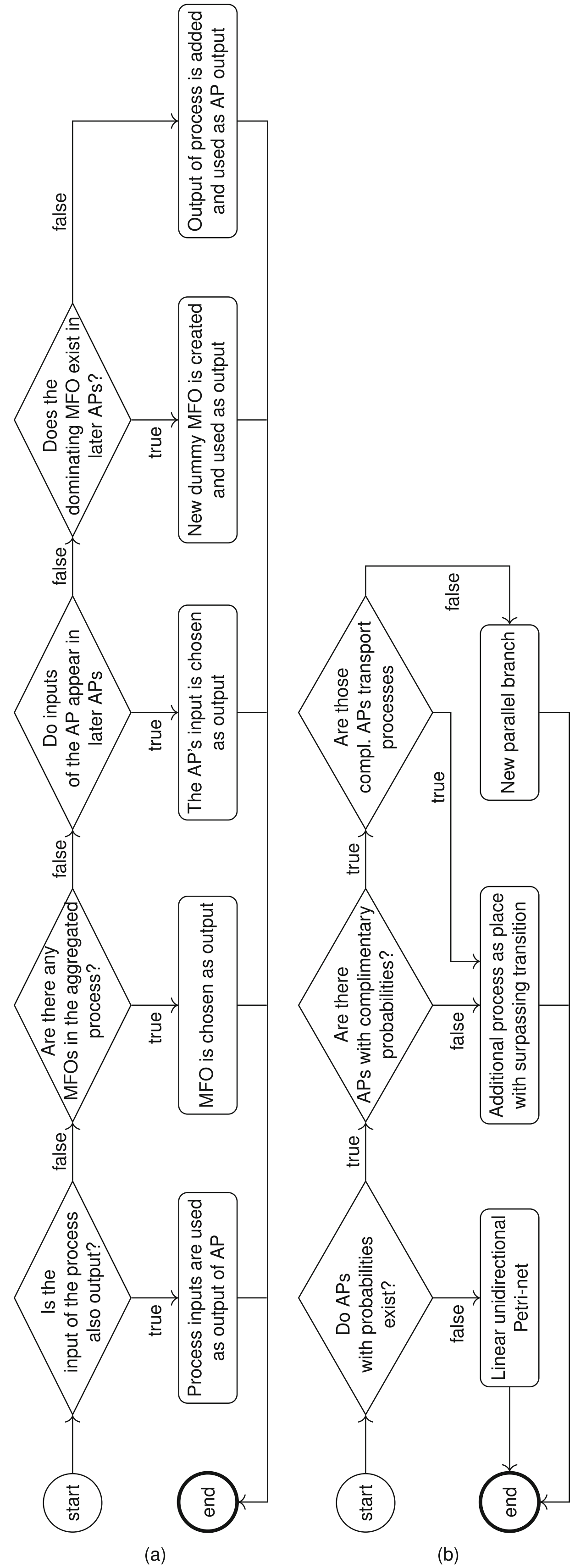

start
Is the input of the process also output?
true
Process inputs are used as output of AP
false
Are there any MFOs in the aggregated process?
true
MFO is chosen as output
false
Do inputs of the AP appear in later APs
true
The AP's input is chosen as output
false
Does the dominating MFO exist in later APs?
true
New dummy MFO is created and used as output
false
Output of process is added and used as AP output
end
(a)
start
Do APs with probabilities exist?
false
Linear unidirectional Petri-net
true
Are there APs with complimentary probabilities?
false
Additional process as place with surpassing transition
true
Are those compl. APs transport processes
true
false
New parallel branch
end
(b)

Fig. 7 Conditional logic for defining the process model as Petri-net

Algorithm 1: Procedure for the analysis phase

```
for each main process ($p$) in project do
    for each process ($p'$) in main process do
        for each sub-process ($p''$) in process do
            Init:
              $n \leftarrow 1$;
              $N \leftarrow$ number of process steps ($p'''$) in sub-process;
              aggregatedProcesses = new Collection();
              while $n = N$ do
                Init:
                  aggregatedProcess $AP$ = new Object;
                  while $\boldsymbol{R}(p'''(n)) = \boldsymbol{R}(p'''(n-1))$ & $n \neq N$ do
                      Add $p'''(n)$ to $AP$;
                      Increment $n \leftarrow n + 1$;
                  end
                  if $AP$ contains MTM codes with factors above
                    limit then
                      Classify $AP$ as transport process;
                  end
                  aggregatedProcesses.addItem($AP$);
            end
        end
        for each $AP$ in aggregatedProcesses do
            Apply conditional logic for inputs and outputs
             // see Figure 7
        end
    end
end
```

tions. Figure 8 depicts the running simulation model. Due to agent-wise data collection, it is possible to access detailed information from every simulation element. The data is saved into a text file when the simulation is completed.

Demonstration

To evaluate whether the proposed framework is applicable to support TPL process design, experiments are set up to test the support for significant planning decisions. Those major decisions concern the layout, process design and resource capacity. Due to confidentiality, presenting real-world cases is not viable. Therefore, the experiments use illustrative case studies to showcase applications usability for the given decisions, as it is a common procedure in design science research (Peffers et al. 2012). First, a case study gives insights into applying the different analysis phases of the modelling sub-system. Then another set of case studies shows the application of the simulation and how it can support the significant decision of designing TPL concepts.

Case study for analysis component

Subsection 3.3 describes the analysis component with its two phases. The logic is applied to an exemplary constructed extract of a would-be process. Table 1 shows a simplified fictitious main process (p), called goods receipt, consisting of three processes (p'): `reception pallet`, `reception package` and `storage`. The physical material flow of the main process is characterised by two separate flows which fuse before they enter the storage area. The first flow considers the delivery of palatalised goods by truck. After checking the truck and doing the paperwork, the truck is unloaded. In this last sub-process (p''), 10% of the delivered pallets need an inspection. All process steps before the quality check are performed at the truck loading spaces; afterwards, the pallets are transported to the buffer area, where the quality check of pallets is done. The transport process is identified by MTM Codes `SABAFM` and `SFISF`, whose largest factor of 45 is beyond the system's minimal station distance, see Table 1. Furthermore, the truck unloading sub-process incorporates a divisor, which relates to handling two pallets simultaneously because of their stackability. The other process `reception package` describes the additional material flow, which accepts packages, and transports them to the buffer area to commission them on a pallet. First, a courier is registered with its package in the sub-process `entrance`. In 5% of the cases, damages and shortages are documented. Afterwards, the package is transported to the buffer area and commissioned onto a pallet. The last process `storage` incorporates two sub-processes, first transporting the goods to the destination inside the warehouse and second scanning the goods and the storage location.

All presented processes are subdivided into the APs by checking for resource, worker type and probability uniformity. The transport processes are identified by their specific MTM codes. As the inputs and outputs of the processes are known, the decision tree in Fig. 7 can be applied: the process `reception pallet` inputs trucks and outputs pallets. The first sub-process incorporates just trucks as MFO. Therefore the third conditional logic applies, as the unit truck appears in later APs. The same logic is applied until the last AP `agg_scan_2`, where the fourth conditional logic is falsified with the result that the process output is used as the sub-processes output. The second process `reception package` and its included APs follow a different logic. Both APs `agg_entrance_1` and `agg_entrance_2` follow the third conditional logic and use the unit `package` as input and output. Next, the AP `agg_entrance_3` falls under the second condition: two-thirds of the AP's units are pallets. Therefore it is an MFO. Thus, the inputs are packages, and the outputs are pallets. From then on, all APs inputs and outputs are pallets. The last process `storage` uses the first conditional logic, as the first AP's input is also the output of the process `storage`. With the inputs and outputs of all APs defined, the second phase of the analysis is due. The APs of the process `reception pallet` only the AP `transport_1` is categorised as transport process and therefore is ignored in building Petri-nets. Furthermore, the

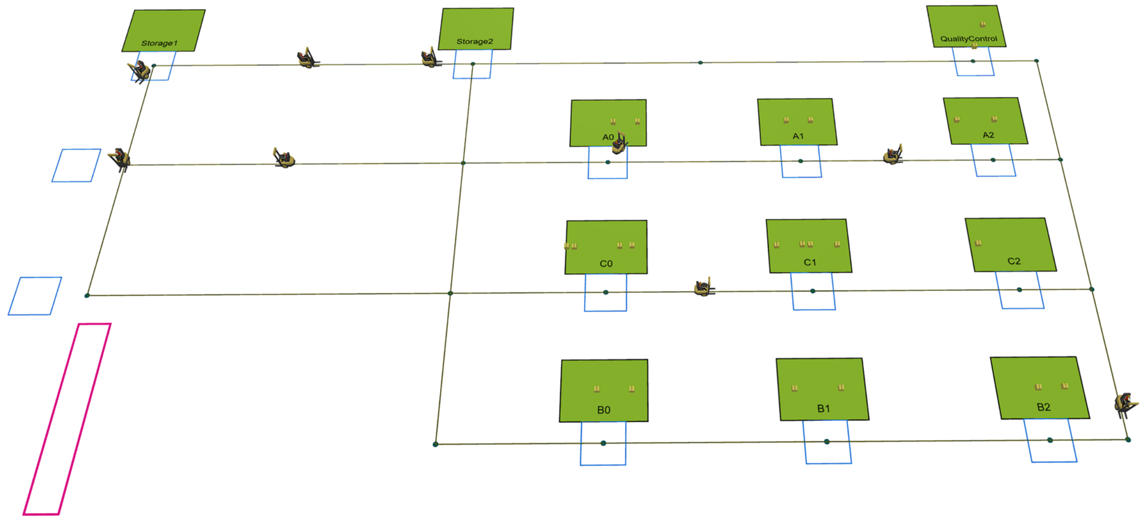


Fig. 8 Running AnyLogic model with a dense layout

second conditional logic applies; the AP agg_quality has a 10% probability with no process with complementary probability. Therefore, it places as an extra process with extra transitions. The same logic applies to the AP agg_entrance_2.

Case studies for planning decisions

Multiple TPL project calculations were used to test the concept and applicability during the underlying research project. As those projects contain highly confidential information on how specific processes are calculated and knowledge of customers' sites, those projects are not eligible to be published. The project planning personnel of TPL providers tested especially alternative layouts and the different numbers of allocated transport resources to see if the demanded performance could be achieved according to the logistic system. To build upon those findings, a selection of influencing parameters is depicted in Table 2, which are then used to construct illustrative case studies for demonstrating the applicability. The described values range wide value spaces to showcase the influence of those values.

Often planning personnel is concerned with finding suitable sites to accommodate logistics operations. The possibilities range from available spaces on the OEMs production site over renting facilities to using leftover space on own sites where operations could be accommodated between existing TPL projects. So configuring alternate layouts for logistic systems and comparing them is crucial for applicability. Therefore, three different layout options were created for *Case Study 1* based on the typical layout in real-world projects; see Fig. 9. The dense layout is used for compact sites, the stretched option mimics a layout based on unidirectional material flow, and the dispersed layout represents typical layouts for brownfield sites. Another critical aspect of TPL projects is resource allocation; this holds for transportation resources and static resources at stations. In the context of the *Case Study 2*, the number of forklifts vary from one to four. In *Case Study 3*, the number of static resources is varied. Specifically, the number of bottleneck resources ranges from two to four. Other resources are not varied to reduce the complexity of the evaluation. Lastly, another exciting aspect when planning TPL services is alterations in processes, e.g. quality problems of the customer can lead to later adjustments of specific processes. In *Case Study 4*, the experiments vary the percentage of products needing quality checks. In *Case Study 5*, cases 1 and 2 are combined to test whether scaled-up transport resources can counteract sub-optimal layouts. Due to detailed data collection during simulation runs, statistics are available. This work shows work in progress (WIP), which defines the number of products in the current system. It indicates whether the system is clogged up and consequently is not logistically able to process the daily program. Further, the product-specific flow times deepens the understanding towards the relation between production rate and resource capacity. To complete this perspective the transport resource utilisation show which impact transport capabilities have. Other statistics could be possible but are left out due to simplicity.

Evaluation

Results

The experiments' results concerning the WIP and flow times are depicted in Figs. 10 and 11. The results of *Case Study 1* are the following: dense and stretched layouts can accom-

Table 2 Experiments

Parameter	Values
Layout	{dense, stretched, dispersed}
Number of Transport Resources	{1, 2, 3, 4}
Number of Static Resources at Bottle Neck Station	{2, 3, 4}
Probability of Quality Process	{5%, 10%, 20%}

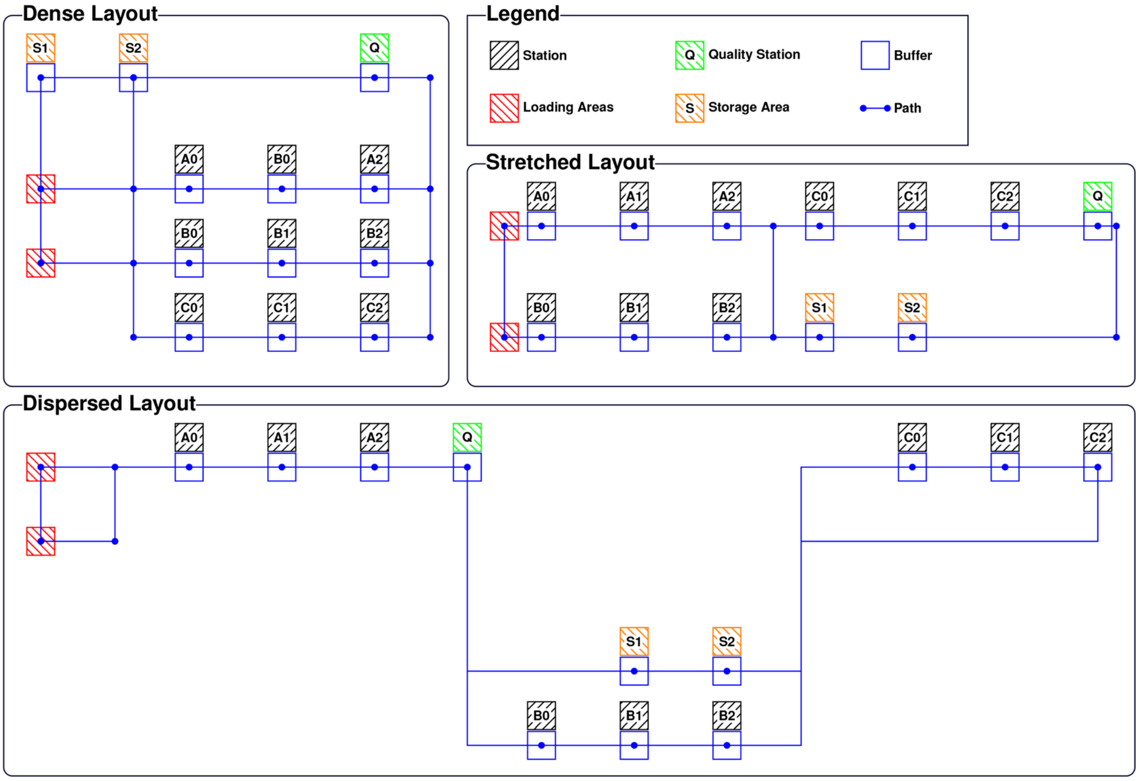


Fig. 9 Different layouts for case studies

plish the daily production programs, and the WIP goes down to zero which is in this case study a defined goal. On the contrary, the dispersed layout cannot reach the set daily goal and empty the shop floor. Furthermore, we see significantly higher flow times of the handled goods. Products in the system with the dispersed layout have up to 50% higher flow times. Similar effects are seen when considering the transport resources' utilisation rates. The dense layout leads to a utilisation rate of 17.0% and the stretched one to 17.4%. In contrast the increased distances between stations lead to a utilisation rate of 24.3%. In *Case Study 2*, the dense layout is chosen for the logistics site. A downside of this site could be a limited space in which only a limited amount of resources can be placed. To counteract this downside, transport resources could be scaled up to utilise the limited resources as much as possible. Thus, there are two bottleneck resources at station C0 combined with the range of possible transport resources. The figures show the strong effect of adding further transport resources. The WIP sets to lower levels with additional forklifts, as the system with four forklifts can process all the goods received in one day. This effect is seen in the presentation of flow times. The lower the number of transport resources, the fewer products are finished. A contrary effect is seen for the utilisation rate. Reducing the transport resources from four to one increases the utilisation rates from 14.8% to 39.6%. In *Case Study 3*, the bottleneck resources are scaled up. The WIP of the logistical system with two bottleneck resources does not reach zero during the shifts of one day. With increasing resources, WIP lowers significantly, especially at the end of a working day. Even though four resources do finish the daily logistical program earliest, three resources also complete the program comfortably. This effect is also visible when considering the flow time; the fourth resource does not add the same benefit in terms of flow time as the third one at station C0. As transport resources are not scaled within the same layout, no actual diversion of utilisation is observed. The rates are between 17.0% and 18.0%. *Case Study 4* considers the change in the quality processes. But in contrast to prior cases, the difference in the probability of a necessary quality check does not influence the logistical capabilities of the logistic system. Figure 11 shows that the WIP does not differ between the different parameterisations of quality checks. The same effect can be seen in the flow time statistics. Furthermore, additional transports to and from the quality checking station do not increase utilisation rates as those are between 17.2% and 18.0%. The fifth and last case builds upon *Case Study 1* and *Case Study 2*. As already seen, the dispersed layout is inferior to the stretched layout. Nevertheless, it is possible that, for example, a worse option is chosen due to the availability of sites. In this case, it is interesting to know if the increased number of transporters can compensate for the more considerable distances between stations in the dispersed layout. The stretched layout with one transportation resource is depicted as the benchmark. The dispersed layout is parameterised with up to three forklifts. It becomes apparent that the dispersed system with two forklifts nearly compensates for the more considerable distances and with three forklifts even surpass it. With increasing transport resources, utilisation rates for the dispersed layouts drop from 59.8% to 23.9%.

Discussion

The targeted functionalities could be proven by applying the developed algorithms and systems to the case studies. The case study for the modelling application demonstrates how the proposed Algorithm 1 performs in an illustrative planning case which is significantly reduced in complexity compared with real-world planning cases. While the overall algorithm and its structure have worked for all projects, its conditional logic in Fig. 7 showed in rare cases difficulties when planners created process structures in ways which calculate correctly to price positions for the tender but do not represent actual process structure. Furthermore, sometimes the parameters of an MTM step are placed incorrectly; for example, information which concerns the divisor is filled in the probability. E.g., the MFO is defined as a package set with another package on a pallet. An AP which needs to be done only once per pallet could be indicated incorrectly by a 50% probability or correctly with a divisor of 2. From the perspective of calculating price positions, the result is the same, but logically, using probabilities is incorrect in this case. So individual wrong behaviour leads to manual rework of the Petri-net model. The given case studies for planning decisions demonstrated the system's capability of simulating different scenarios without changing the underlying simulation environment. Adjusting the model with the component `modelling` allows the user to test different layout, resource and process configurations without specific knowledge of the chosen simulation software. With the presented metrics of WIP, flow time and utilisation rate, first assessments are easy to generate. Nevertheless, the proposed system does not impede manual modelling efforts completely. As modelling mistakes by wrongfully planned processes lead to manual adjustments and layout specifications are not generated from CAD files, users still need to operate the subsystem `modelling`. By testing this model on real-world projects, it became apparent that this system enables the planning personnel to verify the logistical feasibility of planned projects and compare them. Thus, the business process of planning TPL projects is not directly reduced, but new frontiers in evaluating quality are opened. Given costly project audits, determining if planned projects divert from their realised operation, the evaluation through simulated material flow could reduce the difference between those. In situations where dynamic effects influence the performance of a logistics system, such evaluation steps could be crucial. For example, if deliveries of certain goods are delayed or peak transportation demand in the system is underestimated due to the accumulation of arriving deliveries and deliveries to be dispatched, leading to flow times above the customer's demand. The capability of verifying planned projects, which could improve planning quality, is shown. This quality aspect in TPL projects is also addressed by Spiegel et al. (2014) and Veigt et al. (2022). Though Spiegel et al. (2014) used standardisation to reduce planning efforts primarily; it could also incline a standardised quality. Veigt et al. (2022) uses statistical methods and supervised machine learning to propose MTM steps for inexperienced planning personnel. Both works are effective regarding their specific research goal but fall short when improving quality with the logistic performance in mind. To categorise this system into the existing vital factors presented by Reinhardt et al. (2019), the data source is ultimately defined by user input. The spreadsheet tool used the planning personnel to calculate the project, or as Veigt et al. (2022) did it by artificial intelligence, the project's data is filled out by planning personnel. The data variability is static to the extent that live data from operational systems are not fed into the system. But it is dynamic regarding rapid update cycles during the planning process. Information retrieval for the actual simulation model is explicit. Still, taking the subsystem *modelling* into account, the data retrieval is implicit as the spreadsheet could be seen as a knowledge base that, combined with logical reasoning, generates the simulation model's configuration file. Lastly, the standardisation is based on the CMSD but tailored to TPL systems.

Conclusion and outlook

This paper presents a system to support the Third-Party Logistics project planning process by automatically generating material flow simulation models to evaluate planned projects. This includes an ontology for resources, layouts,

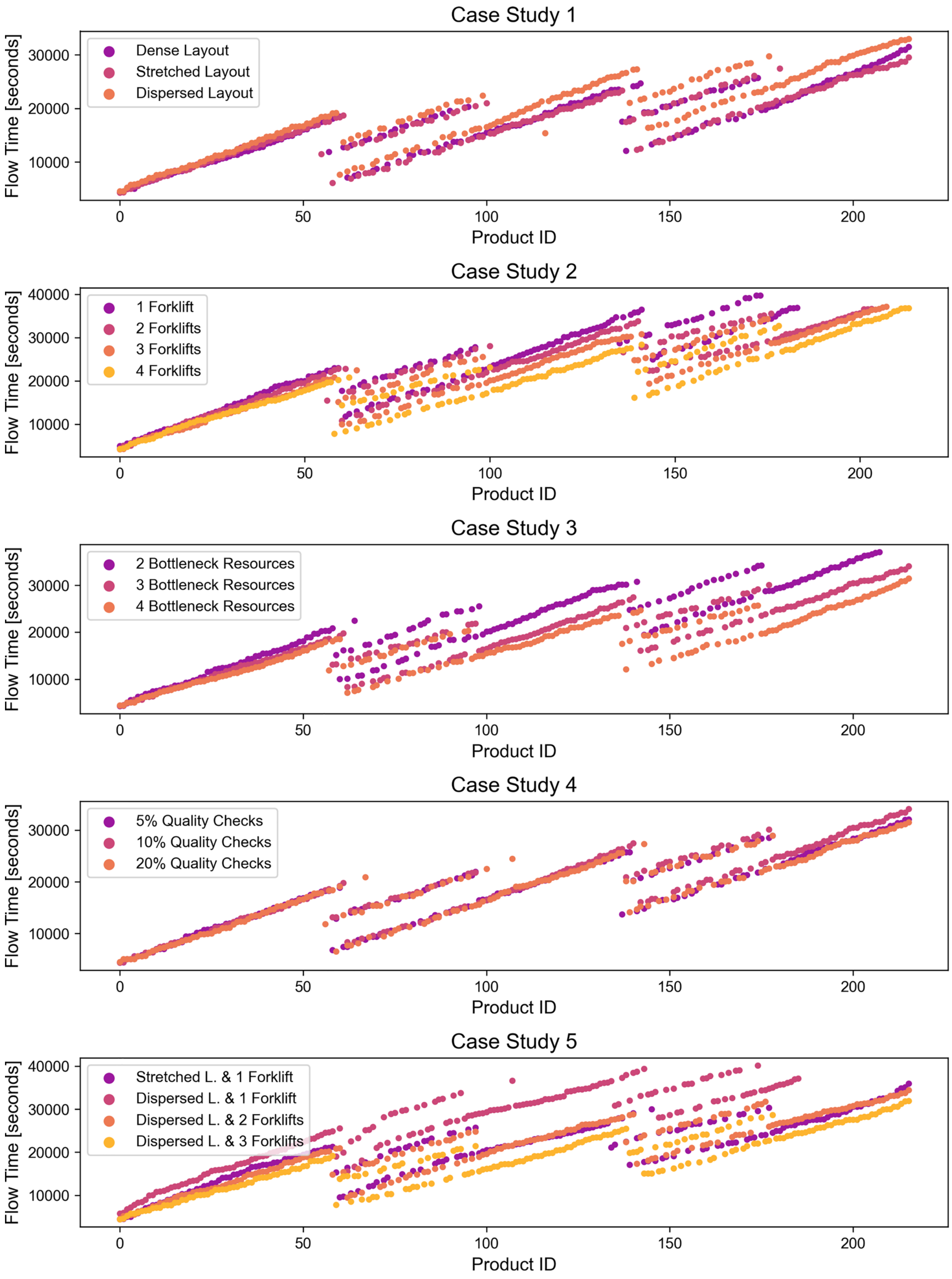


Fig. 10 Flow time results for the five case studies and their' different parameterisation

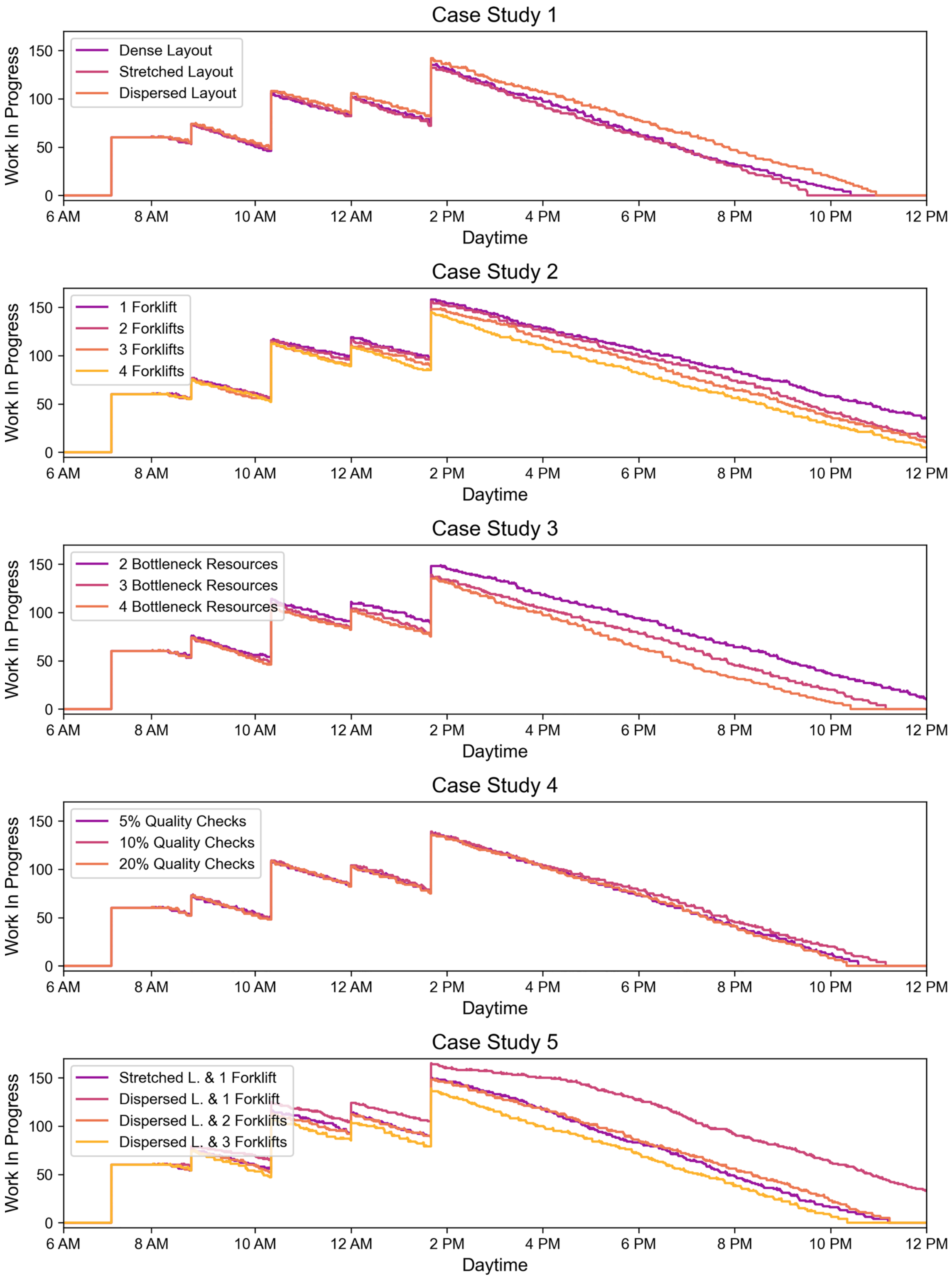


Fig. 11 WIP results for the five case studies and their' different parameterisation

and operational and organisational elements, found in TPL systems. This ontology is used by a proposed system which imports information from spreadsheets used in the calculation phase of TPL projects. After analysing and editing this spreadsheet, it performs a model transformation. The resulting configuration file is then loaded into the ASMG. There, material flow simulations generate metrics that can then be used to evaluate the planned projects, which can be used to adjust the planned project to improve logistic performance. Several case studies have shown the system's applicability to gaining insights into the logistical performance of TPL systems. Usual planning decisions in TPL be performed on illustrative case studies. Those include comparing possible system layouts, scaling up resources, adjusting process models and evaluating resource scale-ups to counteract less efficient layouts. Applying the system to those case studies shows its applicability in TPL systems. Nevertheless, limitations exist. In the current state of the proposed system, longer run times would need a performance-enhancing improvement. Furthermore, the proposed ontology is not directly transformable with standard model transformation techniques. Aside from addressing these issues, a possible extension to the current research is integrating data from TPL operations. Currently, only data acquisition of this system is based on the human experience, so combining a quantitative feedback loop concerning the correctness of planned operations is still to be defined. Therefore, this work could be extended by integration technologies like process mining, whose results are used to optimise TPL concept design processes.

Author Contributions Conceptualization: L. M. Steinbacher, M. Veigt; Methodology: L. M. Steinbacher; Formal analysis and investigation: L. M. Steinbacher; Data Curation: L. M. Steinbacher; Software: L. M. Steinbacher, T. Düe; Writing - original draft preparation: L. M. Steinbacher; Writing - review and editing: L. M. Steinbacher, M. Freitag; Funding acquisition: M. Freitag; Resources: M. Freitag; Supervision: M. Freitag.

Funding This research has been funded by the European Regional Development Fund (ERDF) and the Bremer Aufbau-Bank (BAB) as part of the project "INSERT- AI-based assistance system for concept planning in production and logistics" (project number FUE0626B).

Data availability The data from experiments and models will be made available on request.

Declarations

Conflict of interests The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

Ait Alla, A., Kreutz, M., Rippel, D., Lütjen, M., & Freitag, M. (2020). Simulatedbased methodology for the interface configuration of cyber-physical production systems. *International Journal of Production Research, 58*(17), 5388–5403. https://doi.org/10.1080/00207543.2020.1778209

Barlas, P., & Heavey, C. (2016). Automation of input data to discrete event simulation for manufacturing: A review. *International Journal of Modeling, Simulation, and Scientific Computing, 7*(01), 1630001. https://doi.org/10.1142/S1793962316300016

Bergmann, S. (2014). *Automatische generierung adaptiver modelle zur simulation von produktionssystemen (Unpublished doctoral dissertation)* (p. 2013). Ilmenau: Technische Universität Ilmenau, Diss.

Bergmann, S., & Straßburger, S. (2015). On the use of the core manufacturing simulation data (cmsd) standard: experiences and recommendations. Universitätsbibliothek Ilmenau.

Centobelli, P., Cerchione, R., Murino, T., & Gallo, M. (2016). Layout and material flow optimization in digital factory. *International journal of simulation modelling, 15*(2), 223–235.

Eberlein, S., & Freitag, M. (2022). A pull principle for the material supply of lowvolume mixed-model assembly lines. *Procedia CIRP, 107*, 1385–1390. https://doi.org/10.1016/j.procir.2022.05.162

Fragapane, G., Zhang, C., Sgarbossa, F., & Strandhagen, J. O. (2019). An agent-based simulation approach to model hospital logistics. *International Journal of Simulation Modelling, 18*(4), 654–665. https://doi.org/10.2507/IJSIMM18(4)497

Gaspari, L., Colucci, L., Butzer, S., Colledani, M., & Steinhilper, R. (2017). Modularization in material flow simulation for managing production releases in remanufacturing. *Journal of Remanufacturing, 7*(2), 139–157. https://doi.org/10.1007/s13243-017-0037-3

Group, C.M.S.D.P.D. (2010a). Standard for: Core manufacturing simulation data - uml model. Simulation Interoperability Standards Organization.

Group, C.M.S.D.P.D. (2010b). Standard for core manufacturing simulation data - xml representation. Simulation Interoperability Standards Organization.

Hoellthaler, G., Schreiber, M., Vernickel, K., be Isa, J., Fischer, J., Weinert, N., & Braunreuther, S. (2019). Reconfiguration of production systems using optimization and material flow simulation. *Procedia CIRP, 81*, 133–138. https://doi.org/10.1016/j.procir.2019.03.024

Hoff-Hoffmeyer-Zlotnik, M., Schukraft, S., Werthmann, D., Oelker, S., & Freitag, M. (2017). Interactive planning and control for finished vehicle logistics. Digitalization in Maritime and Sustainable Logistics: City Logistics, Port Logistics and Sustainable Supply Chain Management in the Digital Age. Proceedings of the Hamburg International Conference of Logistics (HICL), vol. 24 (pp. 77–93).

Hofmann, E., & Osterwalder, F. (2017). Third-party logistics providers in the digital age: Towards a new competitive arena? *Logistics, 1*(2), 9. https://doi.org/10.3390/logistics1020009

Jurasky, W., Moder, P., Milde, M., Ehm, H., Reinhart, G. (2021, 10). Transformation of semantic knowledge into simulation-based decision support. Robotics and Computer-Integrated Manufacturing, 71, 102174. https://doi.org/10.1016/j.rcim.2021.102174

Large, R. O. (2011). Partner-specific adaptations, performance, satisfaction, and loyalty in third-party logistics relationships. *Logistics Research, 3*(1), 37–47. https://doi.org/10.1007/s12159-011-0047-8

Lugaresi, G., & Matta, A. (2020). Generation and tuning of discrete event simulation models for manufacturing applications. 2020 winter simulation conference (wsc) (p. 2707–2718).

Lütjen, M., & Rippel, D. (2015). GRAMOSA framework for graphical modelling and simulation-based analysis of complex production processes. *The International Journal of Advanced Manufacturing Technology, 81*, 171–181. https://doi.org/10.1007/s00170-015-7037-y

Milde, M., & Reinhart, G. (2019). Automated model development and parametrization of material flow simulations. 2019 winter simulation conference (wsc) (p. 2166–2177).

Müller, M., Mielke, J., Pavlovskyi, Y., Pape, A., Masik, S., Reggelin, T., & Häberer, S. (2021). Real-time combination of material flow simulation, digital twins of manufacturing cells, an agv and a mixed-reality application. *Procedia CIRP, 104*, 1607–1612. https://doi.org/10.1016/j.procir.2021.11.271

Peffers, K., Rothenberger, M., & Kuechler, B. (Eds.). (2012). *Design science research in information systems advances in theory and practice*. Berlin Heidelberg: Springer.

Peters, M., Cooper, J., Lieb, R. C., & Randall, H. L. (1998). The third-party logistics industry in europe: Provider perspectives on the industry's current status and future prospects. *International Journal of Logistics Research and Applications, 1*(1), 9–25. https://doi.org/10.1080/13675569808962035

Premkumar, P., Gopinath, S., & Mateen, A. (2021). Trends in third party logistics - the past, the present & the future. *International Journal of Logistics Research and Applications, 24*(6), 551–580. https://doi.org/10.1080/13675567.2020.1782863

Rajesh, R., Pugazhendhi, S., Ganesh, K. (2011). Towards taxonomy architecture of knowledge management for third-party logistics service provider. Benchmarking: An International Journal, 18(1), 42–68, https://doi.org/Towardstaxonomyarchitectureofknowledgemanagementforthird-partylogisticsserviceprovider

Reinhardt, H., Weber, M., & Putz, M. (2019). A survey on automatic model generation for material flow simulation in discrete manufacturing. *Procedia CIRP, 81*, 121–126. https://doi.org/10.1016/j.procir.2019.03.022

Ristić, M., & Davidović, J. (2019). The tender for procurement of logistics services and development of logistic partnership. 4th Logistics International Conference

Shariatzadeh, N., Sivard, G., & Chen, D. (2012). Software evaluation criteria for rapid factory layout planning, design and simulation. *Procedia CIRP, 3*, 299–304. https://doi.org/10.1016/j.procir.2012.07.052

Spiegel, T., Siegmann, J., & Durach, C. F. (2014). Flexible development and calculation of contract logistics services. *World Academy of Science, Engineering and Technology International Journal of Industrial and Manufacturing Engineering.* https://doi.org/10.5281/zenodo.1091196

Steinbacher, L. M., Ait-Alla, A., Rippel, D., Düe, T., & Freitag, M. (2022). Modelling framework for reinforcement learning based scheduling applications. *IFACPapersOnLine, 55*(10), 67–72. https://doi.org/10.1016/j.ifacol.2022.09.369

Straube, F., Ouyeder, J., Siegmann, J., & Spiegel, T. (2011). Empirical analysis of the tender management process of contract logistics service providers. In T. Blecker (Ed.), *Maritime logistics in the global economy.* Eul: Lohmar and Köln.

Uhlemann, T.H.-J., Lehmann, C., & Steinhilper, R. (2017). The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia Cirp, 61*, 335–340. https://doi.org/10.1007/s13243-017-0037-3

Veigt, M., Steinbacher, L., Freitag, M. (2021). Planungsassistenz in der Kontraktlogistik - Ein Konzept zur KI-basierten Planungsunterstützung innerhalb einer digitalen Plattform. Industrie 4.0 Management, 37(5), 11–15,

Veigt, M., Steinbacher, L., Freitag, M. (2022). Using supervised learning to predict process steps for process planning of third-party logistics. Dynamics in logistics: Proceedings of the 8th international conference LDIC 2022, bremen, germany (pp. 423–434).

Vernickel, K., Brunner, L., Hoellthaler, G., Sansivieri, G., Härdtlein, C., Trauner, L., & Berg, J. (2020). Machine-learning-based approach for parameterizing material flow simulation models. *Procedia CIRP, 93*, 407–412. https://doi.org/10.1016/j.procir.2020.04.018

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.