A Federated Simulation Framework for Cross-Organisational Processes

Rushan Arshad, Paul Ton de Vrieze, Lai Xu

Arshad R., de Vrieze P.T., Xu L. (2021) A Federated Simulation Framework for Cross-organisational Processes. In: Camarinha-Matos L.M., Boucher X., Afsarmanesh H. (eds) Smart and Sustainable Collaborative Networks 4.0. PRO-VE 2021. IFIP Advances in Information and Communication Technology, vol 629. Springer, Cham. https://doi.org/10.1007/978-3-030-85969-5_24

A Federated Simulation Framework for Cross-Organisational Processes

Rushan Arshad, Paul Ton de Vrieze [1], Lai Xu [1]

[1] Department of Computing and lnformatics, Faculty of Science and Technology
Bournemouth University, Talbot Campus,
Bournemouth, UK, BH12 5BB.
{arshadr, pdvrieze,lxu)@bournemouth.ac.uk

Abstract. Simulation is a significant tool that can be used to evaluate, monitor and enhance the processes and to predict the behaviour of a system in a particular scenario. Collaborative processes involving multiple organisations are becoming important in the changing landscape of the manufacturing industry towards industry 4.0. Simulating these processes require an independent and distributed execution because of the privacy concerns of partner organisations and the reusability of existing simulators. In this paper, we propose a simulation framework based on a federated approach for the simulation of collaborative processes. The federated approach enables the simulation of parts of the processes from multiple organisations by combining independent simulators through a common interface. The common interface is responsible for the synchronisation of all the simulators within the federation. The framework will be evaluated using an industrial case study of textile manufacturing using Virtual Organisations.

Keywords: Industry 4.0, Simulation, Federated Simulation, Collaborative Processes

1 Introduction

The modern industry 4.0 enabled landscape provides for a rich and complex organisation of business. For some products, the design may be done by one organisation, and the production by another, where distribution and marketing are managed by yet two other organisations. At the same time, the customer's experience should be of the same quality as if interacting with only one organisation. While the resulting collaborative networks may be able to handle dynamic market conditions better, the networks themselves are more complex to understand. This is exacerbated by the independence of the organisations making up the collaboration.

Where monitoring of key performance indicators is a key tool in the management of processes [1], it is only a post-hoc tool. Instead, simulation of the processes can allow for a prediction of the indicators ahead of time. The use of simulation tools takes various forms and complexities, from anomaly detection to a 'what-if' analysis [2, 3]. All without disrupting the actual system that is in place.

In the case of industrial systems, simulation requires specialist knowledge that is sometimes only available to the manufacturer of the devices involved. In other cases, the simulation is provided in relation to a Manufacturing Execution System (MES) that coordinates the manufacturing process. When looking at the broader business, the business processes surrounding manufacturing would also need to be simulated. Such simulation is not provided by a machine manufacturer or the MES. Given the complexity and variability involved in managing the different simulation models, it is not realistic to have this simulation done in a monolithic way. Instead, multiple simulators are likely to need to cooperate in simulating business processes involving manufacturing aspects.

Cross-organisational processes introduce an increased complexity. In addition to this, the desire to keep a local control of processes would likely increase the frequency of change to the processes. Overall, this leads to processes that are harder to make, keep correct and be optimised. Simulation can help in addressing this complexity and variability by identifying any errors and anomalies before deployment. After deployment, a comparison of the simulation results with actual performance can be used to identify potential process issues.

Cross-organisational processes involve independent actors with independent, but integrated, processes. To be able to accurately simulate such integrated processes, it is important that the simulation is able to simulate the integrated processes in addition to the integration. At the same time, for various reasons (technical or business), it is desirable or unavoidable to have processes simulated independently. This combines with the need to have multiple simulators for the different process aspects for individual organisations. The solution to both issues is to use a federated simulation approach that allows for coordinating simulators to simulate the integrated outcomes in parallel with the integration of the actual processes.

For example, in the case of a just-in-time production chain, including a supplier of parts, an end manufacturer and a shipping provider, various processes would be involved in the production of a single end product. In the case of a sudden surge in the demand of the end product, the ability to produce the products is (also) limited by the production of the part, as well as by the shipping considerations. In part production, the pure production line capacity comes into play, but also staffing, maintenance and supply considerations. Overall, to determine how the potential increase in production could be realised and with what time frame would require simulating the business and manufacturing processes of all three parties in a way that mirrors the coordination present in the actual production process.

Federated simulation has been used in various contexts, and, in particular, it has been explored by the US armed forces in a military context [4]. A later example can be found [5] in the context of multi-modal transportation. While this clearly shows that federated simulation is feasible and valuable, the work is limited in genericity. In this paper, we address this by proposing, from the context of collaborative industry 4.0 processes, a generic framework for federated simulation.

In contrast to the existing approaches, the proposed framework is capable of simulating collaborative processes involving multiple organisations using existing simulators. The simulators are a part of a federation and a simulation coordinator is used to synchronise and facilitate the communication between the simulators. This enables interoperability while also maintaining the maximum confidentiality of the data being shared between the simulators.

2 Related Work and Simulation Requirements

The design of the federated simulation framework requires an understanding of simulation approaches (especially when applied in industry and business process contexts); the purposes of the simulation; collaborative manufacturing; and existing federated simulation approaches. These will be discussed below.

2.1 Approaches to Simulation

Simulation is used for various purposes. Depending on these purposes, different techniques to simulation are the most effective. For example, physical processes, such as weather prediction, are often best simulated using *System Dynamics*. For other problems, techniques, such as Discrete Event Simulation (DES), Discrete Time Step (DTS) and Agent Based Techniques (ABS), are used.

DES simulates a system based on discrete events that occur at different time intervals (which can vary for each event), whereas, in DTS, the time interval is fixed. On the other hand, ABS consist of Agents which are programmed to do specific tasks by modelling their behaviour. Agents can also interact with other systems and can respond to the dynamic changes to their environment [6].

In both the manufacturing process and business process contexts, the most commonly used technique is Discrete Event Simulation. In manufacturing it is used in almost every stage, starting from facility design and general system design [7, 8] to the material handling stage [9, 10]. As DES is fundamentally a detail-oriented simulation paradigm, it has also been used for operational scheduling (resources, tasks) [11].

Discrete Event Simulation differs from the other techniques in the fundamental way that it is based upon a sequential processing of events in the simulation context. In contrast, the other approaches tend to use computation resources linearly with the simulated time duration. As such, discrete event simulation can be more efficient where it is appropriate. More significantly, the other approaches can, with some restrictions, be mapped to allow for integration with an event-based approach.

2.2 Simulation Evaluation

Simulation models and frameworks are developed to model a system's behaviour and to predict the performance of a system in a specific scenario. Simulation results and their analysis determine how a system is expected to perform in a particular point in time. Hence, the accuracy of such a simulation model is significant.

A simulation model or framework answers particular questions about a problem or application. The purpose of the evaluation or validation is to find out whether or not the simulation model is capable of answering these questions with a reasonable accuracy (which should be determined prior to the development of the model). If the simulation model answers the questions reasonably accurately, then it is said to be acceptable.

Expert opinions can be used from a third party, called Independent Verification and Validation (IV and V), to evaluate a simulation model. The models are evaluated under different parameters by the independent experts. A simulation framework is then applied on the case studies from industry to evaluate its accuracy and applications in real-world scenarios. Usually, a diverse set of case studies is used from different backgrounds to make the evaluation accurate [12].

Another approach being used in the evaluation of simulations is the structural walkthrough. In a structural walkthrough traces of events and states in a specific use case are also used to show that the logic and structure of a framework are valid. The logical and structured walkthrough of a conceptual model consists of a formal explanation and field experts can then check the model correctness. The traces of a conceptual model depict the step-by-step process of the execution and then the correctness of the logic is determined [12].

There are other validation techniques like computerised model verification, operational validity, comparison with other models, statistical validation (type I and type II errors), predictive validation and the Turing Test etc. [12].

2.3 Simulation in Collaborative Manufacturing

In terms of collaborative manufacturing, simulation is used in scheduling in order to optimise production schedules and resource utilization [13]. Supply chain management involving multiple organizations also uses simulation to enhance the production and delivery times [14] and to predict the behaviour of the system under varying demands. DES is commonly used to simulate such systems and these simulations are mostly used to optimise specific parts of the processes, for example production lead time, resource cost etc., and does not entirely focus on a complete simulation of parts of the processes that are involving multiple organisations [15].

In collaborative manufacturing, the coordination of the parts of the processes with time is significant because, otherwise, constituent simulators would be running at different times – any communication about the state or events would be invalid and the data and operations would be inconsistent. Traditional (non-federated) simulations do not focus on the coordination mechanism that helps in enhancing the communication and integration of various parts of the processes. This integration results in improving the accuracy of the simulation and the enhancement of the processes.

The requirements for simulation in different organisations can vary. For example, to simulate a specific part of a collaborative process, details, such as the resource cost, are not required but in another part of the process, it is necessary to simulate the resource cost. Therefore, different simulators with multiple simulation techniques must be used. Moreover, the data that is being used by multiple organisations can be heterogeneous, and thus it would not be feasible for a monolithic simulation model to incorporate this data.

Traditional simulation techniques, like DES, Agent Based Simulation (ABS) and System Dynamics (SD) and their applications in the industry, are largely based upon monolithic models. As such, they have inherent limitations when it comes to cross-organisational processes. The different simulation models need to be integrated in such a way that simulators from different organisational boundaries can communicate with each other and share results so that the overall simulation becomes reliable and complete.

2.4 Existing Approaches in Federated Simulation

Federated Simulation enables the combination of more than one simulation model and incorporates the feedback loops of simultaneously executed simulations [5]. It also enables the communication of operational characteristics and functions of one model with another where there is a probability of any dependency, making the overall simulation more accurate. Federated simulation is preferred because existing simulators can be used to simulate different types of functions/processes.

High Level Architecture (HLA) is a standard [16] that has been developed for simulation interoperability between different simulators by the US Military [4]. HLA consists of some basic rules that govern the interaction of the components of the HLA federation. The components include simulators (federates) and the interface that is responsible for an efficient communication between the federates.

The HLA allows different simulators to be combined in a federation where each simulator has its own data and configurations. A common interface is used to provide communication between these simulators to achieve a simulation objective. For example, multiple simulators using discrete event and discrete time step simulation are combined in a federation to simulate a transportation system [5].

One of the goals of the framework is to support the validation of processes through simulation. Where the processes using a single-instance are long-lived and complex, rather than those using many small instances, and they require simulating processes using a monte-carlo simulation approach on the level of the federation (not only for individual simulators). There are many ways in which component simulators can be adjusted, requiring the framework to be designed accordingly

3 Simulation Framework

A generic framework based on federated simulation (Figure 1) consists of a federated simulation runtime that includes different simulators as part of a federation. These simulators are equipped to simulate various processes, connected through a generic component which is named the simulation coordinator. Each simulator has a local data normalisation component which handles the data interoperability between the simulators. Each of the components within the federation is provided with initial configurations, and, at the end of a simulation run, the data collation module combines the data from the simulators for a comprehensive analysis for the decision support. The working of individual components of the framework is described in the subsequent parts of this section.

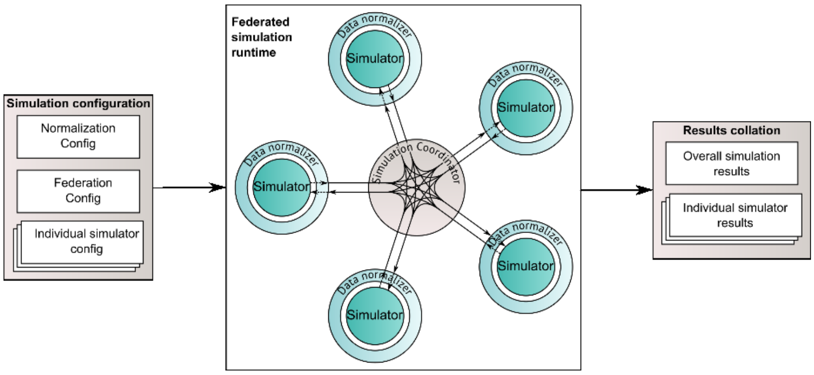

Fig. 1. Conceptual Framework for Federated Simulation

3.1 Top Level Design

Core components of the proposed framework include the initial configuration, simulation coordinator, and the simulators consisting of event and state sharing through the coordinator.

There are three types of configurations: Normalization Configuration, which provides data for the data normalizer to execute its tasks. This is specified through the *initial configuration*. As an example, the common format in which all data needs to be converted before transmission. Federation Configuration provides instructions for the simulation Coordinator whereas individual simulators are also provided with data and instructions to execute their own tasks. These instructions and configurations initiate the simulation process.

The *simulation coordinator* is responsible for the synchronisation of the federated simulators. The queries related to the state of one simulator over another simulator go through the simulation coordinator. Any change in an event or the state of any simulator triggers the simulation coordinator. For example, if a simulator wants to know the status of a resource from the Resource Status simulator (an example simulator), then it sends a query to the simulation coordinator and then the simulation coordinator communicates with the required simulator to get the result (Fig. 1).

Synchronisation is an important part of the simulation framework. Synchronisation is responsible for synchronising the time and state of every simulator within the federation. This helps in maintaining an accurate behaviour of the simulation at a particular time. The synchronisation of simulators enables consistency in the time and state of each simulator and individual simulators are consistent with the time of the coordinator. After the execution of certain events, each simulator time is jumped forward to match the coordinator.

Framework Assumption

Each simulator has its own state and time which can be forwarded to any point in time for its synchronisation.

Algorithm 1 describes how the synchronisation works for the framework given in Figure 1. For each simulator (S), an initial state is initialised as s_i and the time taken for each event to occur is represented by *t*.

S represents the set of simulators. If there is an event(s) *E* available to execute, then the event, which is the earliest in the queue to be executed, will be preferred and the execution of that event will start. After one event, the next event in the queue will be executed. There can be multiple types of events, for example Information Event, Query Event, Notification Event etc. If the event that is to be executed is an Information Event, then the subscribed simulators to that event will be notified and updated with the data from that particular event. Similarly, the simulator generating the query event will be updated with relevant data.

After a certain time T, the simulators are synchronised to the same time by the coordinator. All the simulators within the federation are forwarded to a common point in time. When a process or a part of the processes is executed, the state of the simulator is changed as well. The states of the simulators are also updated after each event.

The synchronisation in a simulation coordinator plays a vital role in the whole simulation scenario. The consistency, accuracy and completeness of a simulation depend on how well the simulators are synchronised; otherwise, the prediction of the working of a system at a particular point in time will not be accurate.

The simulators within the federation *share state and event* data based on the type of communication that is required at a particular time. This data is shared through the simulation coordinator.

The simulators can share the events with each other depending on their respective requirements. *Event sharing* is important in the case where one simulator's execution is dependent on an event from another simulator. A Publish-Subscribe mechanism provides a suitable solution for event sharing because simulators can subscribe to events from a particular simulator based on its requirements. Each simulator has a list of events that it has a subscription for; for example, one simulator is subscribed to all events from another one, whereas it also has a subscription for all the events related to the order of delivery from another simulator.

Algorithm 1: Working of Simulation Coordinator

S: A set of Simulators
T: Current Time
E: Event Types
while True **do**
  $events\colon (T, E), S) = \bigcup_s^S (s.nextEvent, s)$
  if $events = \emptyset$ **then**
    $((nextT, nextE), nextS) = events.firstBy(t \leftarrow ((t, e), s))$
  end if
  $messages = []$;
  for $s \in S$ do
    if $s = nextS$ **then**
      $messages = s.execute(nextE)$
    else
      $s.progressTo(nextT)$
    end if

```
    end for
    for message ∈ messages do
      if message ∈ Queries then
        message.source.queryResult(message.dest.query(message))
      else if message ∈ Notifications then
        message.dest.notify(message)
      else if message is Broadcast then
        for d ∈ destinationsFor(message) do
          d.notify (message)
        end for
      else if message is Subscribe then
        message.dest.subscribe(message)
      end if
    end for
  end while
```

The simulators are also able to *share states* with each other through the simulation coordinator. When a simulator wants to know the state of another simulator, for example, due to the inter-dependencies between them, it can make a request through the simulation coordinator and the resultant state value is provided to the corresponding simulator through the simulation coordinator (Algorithm 1).

Data Normalization supports the exchange of the data between the simulators and also combines the data in a common format (for example XML). Whenever data is transferred from a simulator, it goes through the data normalizer to convert it into a format which is consistent throughout the system. This enables the communication of data between different parts of the federation.

The *results collation* module collects the data from individual simulators and produces the results based on the analysis of the individual as well as the combined reports. The results produced provide support for dynamic scheduling, machine performance and decision support to enhance the processes and system within the industry 4.0 framework.

3.2 Framework Refinements

The proposed simulation framework is equipped with refinements like the publish and subscribe mechanism and cross-simulator resource allocation. These refinements are significant improvements in the implementation of the overall simulation system.

A *publish and subscribe* mechanism is used to share the events and data between the simulators. A simulator can subscribe to a set of events from different simulators based on the requirements. A simulator can also publish the events which are required by the other simulators within the federation.

Allocation of resources is an important part of collaborative manufacturing where resources are being shared by multiple organisations or between different departments within the same organisation. State Object (Vacant or Busy) can be used to allocate resources. If a resource is required, the status of the resource is checked through the simulation coordinator and then the resource is allocated accordingly. A complete simulator for this purpose is not necessary.

4 Evaluation of Simulation Framework

To evaluate the proposed federated simulation framework, a case study from literature is used. Section 4.1 describes this case, after which Section 4.2 applies the framework to the case.

The proposed federated simulation framework is evaluated in this section using an industrial case study derived from the literature. The business case, as well as the details of an application of the simulation framework in context of the use case in consideration, is discussed.

4.1 Description of the Case Study

The case study to evaluate the framework is derived from [17] with some modifications. The case in consideration involves two companies (Company A and Company B for anonymity) which collaborate to deliver an order for thousands of school uniforms. Both companies belong to the textile industry. Company A specialises in women's clothing and fabrics with the state-of-the-art facility for sample production and highly customised products. Company A also has a broader value chain consisting of modelling, design, production and delivery. Company B specialises in generic clothing fabrics (particularly synthetic fibre fabrics) and is one of the largest exporters to the USA.

There are two types of product that are produced. One is Engineered to Order (ETO) and the other is Customised to Order (CTO). ETO is based on specific customer requirements with particular design and production specifics, whereas CTO refers to mass customisation; for example, a type of product ordered by a number of companies.

4.2 Applying Framework to a Case Study

The two companies, Company A and Company B reach an agreement to form a Virtual Organisation (VO) in which each partner has separate responsibilities. Company A has expertise in ETO, and so the highly customised orders are fulfilled by company A and for mass customisation, like the uniform, orders are executed by company B.

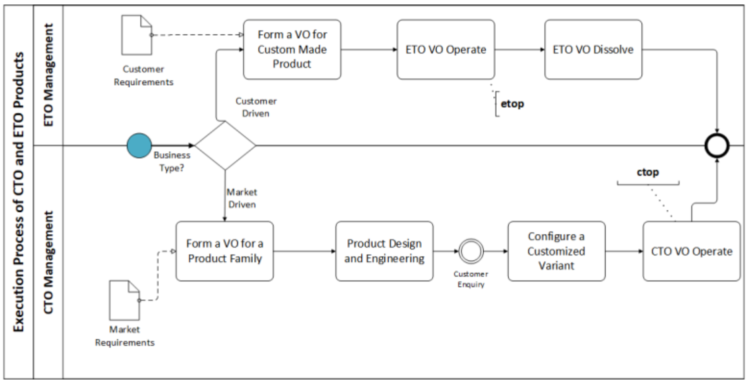

Fig. 2. Case Study Process

Two separate simulators are used within a federated simulation environment to simulate the processes in both companies. One simulator deals with the ETO products and the other simulates the production of the CTO products. The communication between the simulators is done through the simulation coordinator. An example of the process for this case, depicting the execution of CTO and ETO, is depicted by Fig 2.

Events, states and data are shared between the simulators at different stages as required by the processes. For example, data regarding the customisations for a part of the order is shared between the simulators and then when the customisations are finished, the state and events data regarding this process is also shared between the simulators through the coordinator. State transitions within a simulation run are shown in Table 1.

The *events* that are executed by *Actors* are *cbt* (check business type), *ctop* (CTO Process), *etop* (ETO Process), and *start*. The time$\cdot t_i$s is used for time just before *t,* whereas t_is is used for a time just after *t*. Each simulator has its own time and after a certain time period (execution of events), the local time of simulators is synchronised with the coordinator time.

The transitions of state and time in Table 1 show the step by step execution of different parts of processes. Simulator 1 (*s1*) is responsible for executing ETO, whereas CTO processes are executed by Simulator 2 (*s2*). These step by step transitions show that the proposed framework is applicable to the case in consideration.

Table 1 Transition of States

Time C	Events	Actor	Action	time s_1	time s_2
$\cdot t_0$		C	Init	$\cdot t_0$	$\cdot t_0$
$\cdot t_0$	$(t_0, s_1, start), (t_0, s_2, start)$	C	first	$\cdot t_0$	$\cdot t_0$
$\cdot t_0$		s_1	$start$	$t_0 \cdot$	...
$\cdot t_0$	$(t_1, s_1, cbt), (t_0, s_2, start)$	C	first	...	...
$\cdot t_0$		s_2	$start$	...	$t_0 \cdot$
$t_0 \cdot$		C	Sync	...	...
$t_0 \cdot$	$(t_1, s_1, cbt), (t_5, s_2, ctop)$	C	First	$t_0 \cdot$	$t_0 \cdot$
$\cdot t_1$		s_1	cbt	$t_1 \cdot$	...
$t_1 \cdot$	–	C	Sync	...	$t_1 \cdot$
$t_1 \cdot$	$(t_2, s_1, etop), (t_6, s_2, ctop)$	C	First	$t_1 \cdot$	$\cdot t_1$
$t_1 \cdot$	$(t_2, s_1, etop), (t_6, s_2, ctop)$	C	first	$t_1 \cdot$	$\cdot t_1$
$\cdot t_2$		s_1	$etop$	$t_2 \cdot$	$\cdot t_2$

$t_2 \cdot$	–	C	Sync	...	$t_2 \cdot$
$t_2 \cdot$	$(t_3, s_1, cbt), (t_7, s_2, ctop)$	C	First	$t_2 \cdot$	$t_2 \cdot$
$\cdot t_2$		s_1	$etop$	$t_2 \cdot$	$\cdot t_2$
$t_2 \cdot$	–	C	Sync	...	$t_2 \cdot$
$t_2 \cdot$	$(t_3, s_1, cbt), (t_7, s_2, ctop)$	C	First	$t_2 \cdot$	$t_2 \cdot$
$\cdot t_3$		s_1	cbt	$t_3 \cdot$	...
$t_3 \cdot$	–	C	Sync	...	$t_3 \cdot$
$t_3 \cdot$	$(t_4, s_1, etop), (t_8, s_2, ctop)$	C	First	$t_3 \cdot$	$\cdot t_1$
$\cdot t_4$		s_2	$ctop$	$t_3 \cdot$	$t_4 \cdot$
$t_4 \cdot$	–	C	Sync	$t_4 \cdot$	...
$\cdot t_5$		C	End	$\cdot t_5$	$\cdot t_5$

5 Conclusion and Future Challenges

Simulation is a significant tool to detect errors at design time and is used to predict the behaviour of a system at a specific point in time. In the context of modern industrial systems, especially those processes that are involve multiple organisations, simulation becomes more challenging due to the heterogenity of processes and the data involved. We propose a generic simulation framework based on a federated simulation, which allows for simulating different parts of the process in separate but distributed simulators in parallel. This federation helps an organisation to share only the necessary details with other simulators, protecting the confidentiality of the data of the different organisations that are involved in the execution of the processes. A simulation coordinator is responsible for coordinating the data exchanges and the synchronisation of the simulators. An industrial case study of a textile sector was used to demonstrate the function of the working of the framework.

While, overall, the framework is sufficient to support a coordinated simulation there are also limitations. While information sharing can be atomic, based upon a full order of events, the ordering of events is not defined by the coordinator. As such, different simulations of the same configuration could be ordered differently and have different results. The framework can apply various optimisations, in particular, for a repeated simulation of the same scenario. In addition, resources would benefit from special handling.

Acknowledgements
This research is partially funded by the State Key Research and Development Program of China (2017YFE0118700) and it is part of the FIRST project which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 734599.

References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of business process management

2. Vieira, A.A.C.a.D.L.M.S.a.S.M.Y.a.P.G.A.B.a.O.J.A.: Setting an industry 4.0 research and development agenda for simulation – A literature review. International Journal of Simulation Modelling 17, 377--390 (2018)
3. Rodi: Industry 4.0 and the new simulation modelling paradigm. Organizacija 50, 193--207 (2017)
4. Sung, C., Kim, T.G.: Framework for simulation of hybrid systems: Interoperation of discrete event and continuous simulators using HLA/RTI. pp. 1-8. IEEE, 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation (2011)
5. Wall, T.A.a.R.M.O.a.F.R.a.H.M.P.: A federated simulation method for multi-modal transportation systems: combining a discrete event-based logistics simulator and a discrete time-step-based traffic microsimulator. Simulation 91, 148--163 (2015)
6. Bth, L.a.B.N.a.H.C.a.T.S.: Introducing agent-based simulation of manufacturing systems to industrial discrete-event simulation tools. 1141--1146 (2017)
7. Greasley, A.: Using simulation for facility design: A case study. Simulation Modelling Practice and Theory 16, 670--677 (2008)
8. Jagstam, M.a.K.P.: A handbook for integrating discrete event simulation as an aid in conceptual design of manufacturing systems. 2, 1940--1944 (2002)
9. Durieux, S.a.P.H.: Regression metamodeling for the design of automated manufacturing system composed of parallel machines sharing a material handling resource. International journal of production economics 89, 21--30 (2004)
10. Hao, Q.a.S.W.: Implementing a hybrid simulation model for a Kanban-based material handling system. Robotics and Computer-Integrated Manufacturing 24, 635--646 (2008)
11. Koh, S.C.L.a.S.S.M.: MRP-controlled manufacturing environment disturbed by uncertainty. Robotics and Computer-Integrated Manufacturing 19, 157--171 (2003)
12. Sargent, R.G.: Verification and validation of simulation models. Journal of simulation 7, 12--24 (2013)
13. Holweg, M.a.D.S.M.a.H.P.a.N.M.M.: Towards responsive vehicle supply: a simulation-based investigation into automotive scheduling systems. Journal of Operations Management 23, 507--530 (2005)
14. Jung, J.Y.a.B.G.a.P.J.F.a.R.G.V.a.E.D.: A simulation based optimization approach to supply chain management under demand uncertainty. Computers \& chemical engineering 28, 2087--2106 (2004)
15. Liotta, G.a.K.T.a.S.G.: Optimization and simulation of collaborative networks for sustainable production and transportation. IEEE Transactions on Industrial Informatics 12, 417--424 (2014)
16. IEEE Standard for Modeling and Simulation (M S) High Level Architecture (HLA)-- Framework and Rules. IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000) 1--38 (2010)
17. Carneiro, L.a.S.A.H.M.a.A.R.a.A.A.a.F.R.a.F.P.S.: Reference model for collaborative manufacturing of customised products: applications in the fashion industry. Production Planning \& Control 25, 1135--1155 (2014)