Distributed Approaches to Supply Chain Simulation: A Review

NAVONIL MUSTAFEE, Centre for Simulation, Analytics and Modelling (CSAM), Exeter University, Exeter, EX4 4PU, UK
KORINA KATSALIAKI, School of Humanities, Social Sciences and Economics, International Hellenic University, 14th km Thessaloniki-N. Moudania, Thessaloniki, 57001, Greece
SIMON J. E. TAYLOR, Department of Computer Science, Brunel University London, Uxbridge, UB8 3PH, UK

The field of **Supply Chain Management (SCM)** is experiencing rapid strides in the use of Industry 4.0 technologies and the conceptualization of new supply chain configurations for online retail, sustainable and green supply chains, and the Circular Economy. Thus, there is an increasing impetus to use simulation techniques such as discrete-event simulation, agent-based simulation, and hybrid simulation in the context of SCM. In conventional supply chain simulation, the underlying constituents of the system like manufacturing, distribution, retail, and logistics processes are often modelled and executed as a single model. Unlike this conventional approach, a **distributed supply chain simulation (DSCS)** enables the coordinated execution of simulation models using specialist software. To understand the current state-of-the-art of DSCS, this paper presents a methodological review and categorization of literature in DSCS using a framework-based approach. Through a study of over 130 articles, we report on the motivation for using DSCS, the modelling techniques, the underlying distributed computing technologies and middleware, its advantages and a future agenda, and also limitations and trade-offs that may be associated with this approach. The increasing adoption of technologies like Internet-of-Things and Cloud Computing will ensure the availability of both data and models for distributed decision-making, which is likely to enable data-driven DSCS of the future. This review aims to inform organizational stakeholders, simulation researchers and practitioners, distributed systems developers and software vendors, as to the current state-of-the art of DSCS, and which will inform the development of future DSCS using new applied computing approaches.

Q1

CCS Concept: • Computing methodologies~Modeling and simulation~Simulation types and techniques~Distributed simulation • Applied computing~Operations research~Industry and manufacturing~Supply chain management • General and reference~Document types~Surveys and overviews;

Additional Key Words and Phrases: Supply chain management, distributed simulation, simulation methods, review

---

Authors' addresses: N. Mustafee (corresponding author), Centre for Simulation, Analytics and Modelling (CSAM), Exeter University, Exeter, EX4 4PU, UK; email: n.mustafee@exeter.ac.uk; K. Katsaliaki, School of Humanities, Social Sciences and Economics, International Hellenic University, 14th km Thessaloniki-N. Moudania, Thessaloniki, 57001, Greece; email: k.katsaliaki@ihu.edu.gr; S. J. E. Taylor, Department of Computer Science, Brunel University London, Uxbridge, UB8 3PH, UK; email: simon.taylor@brunel.ac.uk.

1049-3301/2021/07-ART25 $15.00
https://doi.org/10.1145/3466170

ACM Reference format:
Navonil Mustafee, Korina Katsaliaki, and Simon J. E. Taylor. 2021. Distributed Approaches to Supply Chain Simulation: A Review. *ACM Trans. Model. Comput. Simul.* 31, 4, Article 25 (July 2021), 31 pages.
https://doi.org/10.1145/3466170

---

1 INTRODUCTION

The shift of management focus from individual businesses to networks of interlinked enterprises, or supply chains, has been evident since the 1990s [Eddama & Coast 2008; Lambert & Cooper 2000]. Supply Chains are multifaceted networks of business entities involving processes from procurement and manufacturing to sales and support [Stevens 1989]. They involve forward and backwards flows of products, information and monetary exchange, which are influenced by variable supply and demand across the supply chain. This is intensified by global markets encouraging hypercompetitive environments. Shortened product lives and increased customer expectations have forced companies to focus their attention on supply chain management [Bandinelli et al. 2006]. Modern **supply chain management (SCM)** approaches favour a global, holistic view in which the individual business entities, or echelons, share information and trust each other, rather than only trying to optimize their local processes independently [Chapman & Corso 2005]. The complexity and dynamic behaviour of supply chains can make SCM challenging. There are different approaches to supply chain design and analysis, including deterministic analytical models, stochastic analytical models, economic models, and simulation [Beamon 1998]. Arguably, simulation is an attractive decision support technique as it can accommodate complex, dynamic behaviour and can be used to explore system sensitivity and explore "what-if" scenarios across a range of key performance indicators and parameters [Mertins et al. 2005; Terzi & Cavalieri 2004; Banks et al. 2002]. Simulation techniques commonly used to analyze supply chains include **discrete-event simulation (DES)** [Law and Kelton 2000], **system dynamics (SD)** [Sterman 2001], **agent-based simulation (ABS)** [Davidsson 2000], Monte Carlo simulation [Jellouli and Chatelet 2001; Mooney 1997], and, more recently, through a combination of hybrid simulation [Brailsford et al. 2019].

A **Supply Chain Simulation**, or **SCS**, is generally implemented as a single computer model. Such models may include processes specific to the various echelons of the supply chain (e.g., manufacturers, distributors and retailers) and include transport and logistics operations. In this conventional modelling approach, decision-makers responsible for the constituent parts of the supply chain will collectively conceptualize the model, provide knowledge of their respective operational processes, provide input data/distributions, co-develop KPI metrics and scenarios for experimentation. This model co-development process thus allows the stakeholders to experimentally explore supply-chain issues such as demand management, production scheduling, product routing, and order fulfilment. The review paper by Terzi & Cavalieri [2004] presents several examples of the application of conventional SCS for demand and sales planning, production planning and scheduling, inventory management, distribution and transportation [Terzi & Cavalieri 2004]. More recently, such models have investigated supply chain-centric business models [Jia et al. 2016]. Distinct from the conventional approach, a supply chain simulation can also consist of multiple models, each of which may represent one or more echelons of the supply chain. Multiple models may become necessary if, for example, the tiers of the supply chain are averse to sharing data and for reuse of verified and validated models. Since the late 1970s, the field of *Parallel and Distributed Simulation* has studied approaches to distributing a simulation across many computers and linking together and reusing existing simulations running on one or more processors [Fujimoto 2000]. Co-ordinated execution of such distributed models over different computers requires specialist distributed computing software. This software is called distributed simulation middleware, and we refer to this simulation approach as a distributed supply chain simulation or DSCS. In the remainder of the paper, the acronym **DS** refers to **Distributed Simulation.** The next paragraph presents a short introduction to the technical aspects of DS.

The middleware for DS implements algorithms that enable the synchronised execution of computer models simulated as separate processes either in the same computer or, more generally, in different computers. Irrespective, the general principle remains the same - the models are independent entities within a DES program, with inter-model communication taking place through distributed computing approaches such as sockets and web services. The algorithms are generally referred to as time management algorithms as their primary function is the co-ordination of simulation time among concurrently executing DES programs. We explain the importance of time management in a DS context by taking the example of a DES program that executes a model using the three-phase approach [Tocher 1963]. This approach is also referred to as the ABC approach and can be implemented by a program which has the following three crucial components - the simulation engine (simulation executive), the logical simulation clock, and a list of events that are maintained in an increasing timestamped order (the event list). The simulation executive loops through the event list and repeats the 'ABC phases of DES' until the end of the simulation. In phase A, it finds the next event from the event list and then advances the simulation clock to the time of the next event; in phase B, it executes all scheduled bound events; in phase C, it tries and executes all conditional events [Tocher 1963]. As every DES taking part in the execution of an overarching DS will have its simulation clock and an event list, the time management algorithms are essential for the prevention of causality errors. Causality errors happen as a result of a failure to process simulation events in increasing timestamp order. More specifically, it occurs when a simulation has processed an event with timestamp T1 and subsequently receives another event with timestamp T2, wherein T1 > T2. Since the execution of the event with timestamp T1 will have normally changed the state variables that the subsequent event with timestamp T2 will use, this would amount to simulating a system in which the future could affect the past [Fujimoto 1990]. For a conventional simulation executed using one DES program, it is fairly easy to avoid a causality error as there is only one logical clock and only one event list. In the case of DS, however, the avoidance of causality is a lot more difficult because it has to deal with multiple event lists and multiple logical clocks.

Our review aims to inform supply chain stakeholders, simulation researchers and practitioners, distributed systems' developers and software vendors, as to the current state-of-the-art of DSCS. This will inform the development of integrated **modelling and simulation (M&S)** solutions for the future supply chains, e.g., symbiotic SCS and hybrid modelling with the combined application of DSCS with new applied computing approaches and Industry 4.0 technologies [Gunal 2019]. Industry 4.0 encompasses several concepts, such as smart factories, cyber-physical systems, self-organization, new approaches to product and service development, and new distribution and procurement systems [Lasi et al. 2014]. Among these, in the context of M&S, the concept of smart factories is arguably the most significant. In an Industry 4.0 smart factory, manufacturing processes are equipped with sensors, autonomous systems and ubiquitous computing technologies, and which enable autonomous control of the production processes through the digitalized models of products and factories [Lasi et al. 2014] - also known as "digital twins". The growth of Industry 4.0 and ubiquitous computing technologies such as the **Industrial Internet-of-Things (IIoT)**, real-time data streaming technologies, Cloud computing, Big Data and analytics and cybersecurity, ensures the availability of data and models for distributed decision making. The data-driven DSCS of the future could be radically different from what has hitherto been reported in the literature, for example, digital twins of supply chain ecosystems, execution over third-party environments to ensure data hiding, Internet/Web-based/'Simulation as a Service' execution of models developed using heterogenous **Commercial, Off-the-shelf (COTS)** simulation packages. Irrespective, the technical literature associated with distributed simulation will be relevant since the data-driven supply chain models of the future, which may be executed over Cloud, High Performance Computing or **General-Purpose Graphics Processing Unit (GPGPU)** environments, will continue to rely on algorithms for time-stamped message exchange and protocols for coordinated execution of models (also referred to as distributed simulation synchronization). This is the realm of the theory of distributed discrete-event simulation [Misra 1986] and its underlying protocols, such as the classic Chandy-Misra-Byrant conservative time synchronization [Chandy and Misra 1981; Bryant 1977] and Jefferson's Time Wrap optimistic algorithm [Jefferson 1985]. Although the fundamentals of distributed simulation date back to the 1970-80s, the topic continues to be relevant today. Recent papers on distributed simulation present new frameworks and algorithms [e.g., Marzolla et al. 2020; Tampouratzis et al. 2020; Li et al. 2019] and report new applications, e.g., distributed simulation for realizing scalable cloud data centres [Elahi et al. 2020], linking data to different model types in the area of industrial construction [Pereira et al. 2020] and cyber-physical co-simulation of shipboard electrical systems based on a distributed simulation standard [Wu et al. 2020].

The remainder of the paper is structured as follows. In Section 2, we compare DSCS with conventional SCS. We take the example of the blood supply chain and discuss both the distributed and the traditional variants of the model and, through this example, highlight the motivation of developing a DSCS. In Section 3, we outline the methodological approach used in identifying papers that constitute our dataset and present the literature analysis framework. Section 4 presents the results from the literature review, followed by discussion in Section 5. Section 6 is the concluding section. It outlines the limitations of this research and a detailed list of suggestions for future work.

2 COMPARING CONVENTIONAL AND DISTRIBUTED SUPPLY CHAIN SIMULATION: AN EXAMPLE OF THE BLOOD SUPPLY CHAIN

In this section, we take the example of an existing work on blood supply chains to perform an informed assessment of conventional and distributed SCS in terms of support for privacy/data hiding, model reusability, execution speed, and so on. This is used as a reference point, since this SCS study has both conventional and distributed implementations in the context of a single case study. More specifically, the study implements both a standard one-computer simulation of a UK blood supply chain [Katsaliaki & Brailsford 2007], as well as its distributed counterpart with several models executed over multiple computers [Mustafee et al. 2009].

A blood supply chain requires processes specific to donor blood collection, testing of blood, central storage, blood distribution, local hospital storage, and so on, to be represented in an SCS model; multiple organizations may be responsible for these processes. For example, in the UK, the ***National Health Service Blood and Transfusion Services (NHS BT)*** is responsible for donor collections, testing and issue of blood. In contrast, individual hospitals are responsible for local policies on ordering, inventory management and transfusion. An example of a conventional simulation is the one developed by Katsaliaki & Brailsford [2007] in which a single SCS model was implemented in a DES environment and was executed on a single computer; the model consisted of the processes of the NHS BT and several other hospitals. Mustafee et al. [2009] extended the conventional model of the NHS BT supply chain and developed a distributed model. It was composed of five separate models, one for the central NHS BT and four other DES models that represented processes specific to individual hospitals. They used IEEE1515 High Level Architecture standard and RTI1.5NG middleware for the execution of the HLA federation. In this work, the motivation to develop a distributed model was the need for *faster simulation execution*. However, there could be other motivations. For example, we may consider the scenario of a private healthcare system where individual hospitals may be reluctant to share their local blood management policies. The

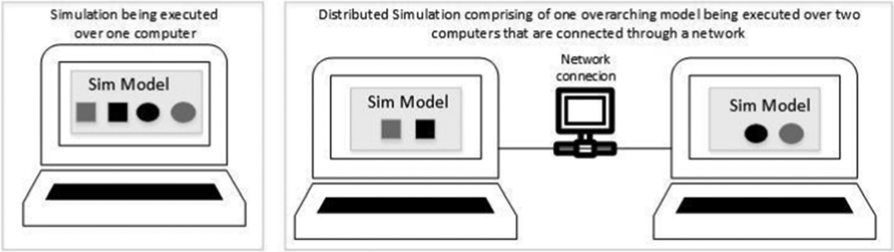

Fig. 1. Conventional simulation is executed over one computer (left). A DS is executed over multiple computers that are connected to a network (right).

supplier may also be unwilling to make available internal process information, as there may exist a market for blood products with more than one supplier in competition for business from different hospitals. In such cases, a single computer model may not be reliable as the interconnected parts of the systems may be unwilling to divulge with the organizational knowledge and data required for implementing an all-encompassing supply chain model. A DSCS alleviates such issues about *privacy/data-hiding* and may *facilitate the reuse of models*; DSCS enables individual entities to exert control over models that represent processes specific to their organizations while being a part of the overarching distributed simulation federation. Thus, DSCS presents new possibilities related to model reuse, privacy, data integrity, and simulation speed. Figure 1 provides an illustration of a standalone simulation and a DS. Table 1 presents a comparison of the conventional one-computer simulation approach with the DS approach.

DS has been widely applied in areas such as defense [Dahmann et al. 1997], combat modelling [Tolk 2012] and in building large-scale network simulations [Taylor et al. 2012a]. However, despite the increasing number of standards, models and architectures which are available for determining how and when a DS can be used for SCS [Bandinelli et al. 2006] and the potential benefits of DSCS (Table 1), there is slow uptake in the industry [Boer et al. 2009; Robinson 2005]. DSCS is complex as multiple models often need to be simulated in different machines in a synchronized manner. The synchronized execution of models necessitates underlying mechanisms for time-stamped message transfers using a point-to-point approach (non-centralized), or, more commonly, by using distributed simulation middleware that handles simulation time advance and information transfer via a computer network. Through this literature review, we identify the DS standards and middleware that have been used in the context of SCS, the different simulation techniques used to model the supply chains, the CSPs and programming languages to implement the models, the DS testbeds to execute the supply chain models, and whether the studies contributed to real-world solution implementation.

3 METHODOLOGY AND FRAMEWORK FOR LITERATURE ANALYSIS

The set of articles for a literature review (subsequently referred to as its dataset) may comprise of *representative work* that is selected by the authors based on their knowledge of the subject area, for example, the review paper on crowd modelling and simulation by Zhou et al. [2010] followed this approach. Articles for the review can also be identified through a two-stage process, wherein a preliminary dataset is first selected through a methodological approach that involves keyword searches using scholarly databases. This is followed by reading the abstracts to establish the relevant articles for the final dataset, for example, Brailsford et al. [2019] employed this two-stage approach in their state-of-the-art review on hybrid simulation. For the DSCS review, we have

Table 1. Comparing Conventional Supply Chain Simulation (SCS) with Distributed Supply Chain Simulation (DSCS)

	Conventional SCS	**DSCS**
Model Reuse	A supply chain is modelled as one single computer model, and it may consist of one or more echelons of the supply chain. The models are often implemented using COTS simulation packages, e.g., Simul$^{\text{TM}}$, Arena$^{\text{TM}}$. It may prove to be challenging to "cut and paste" parts of a model relating to a supply chain entity due to model dependencies. Reuse of models across COTS packages (i.e., to reuse part of a model that has been developed in one simulation software and use it in a different software) is not an option.	Distinct simulations models are developed for individual supply chain entities. The models are often implemented using COTS simulation packages. These are significantly easier to reuse as the "interface" between simulations is relatively simple. Simulations developed in different packages can be linked together as the interface between packages is again relatively simple.
Privacy	A conventional SCS does not allow privacy of data as all features are available to all users. However, some packages may allow hiding source data (e.g. production schedules).	Different organizations simulate the distinct supply chain models within the confines of their firewall (a few sockets are kept open for communication with other models). Model detail and source data are hidden. Users see only a high-level view or just their models and experiment results.
Data Integrity	Single simulations typically use local SQL databases, Excel spreadsheets, flat files, and so on. If the local data sources are not linked to the supply chain databases (which may belong to several organizations), they can be out of date when the simulations are executed. Linking the data sources can be complex, depending on what database technology is used.	As there is no one overarching SCS model, but rather, several sub-models of the underlying supply chain held in different organisations, they can be programmed to use the database sources directly when experiments are being executed.
Speed	They are limited by the hardware specification of the computer running the simulation.	The processing load is shared over several computers running the distributed simulation (communication/synchronization overhead will exist).
Implementation	In the industry, simulation models are generally developed using COTS simulation packages. This is also true for the implementation of conventional SCS models.	In the case of DSCS, a Commercial, off-the-shelf distributed solution does not exist. So, although the individual SC models may be programmed using commercial packages, specialist knowledge of distributed simulation will be required to realize a DSCS. This often requires additional expertise only available from specialist consultancies or research institutions.

Table 2. Literature Analysis Framework using the PPMO Framework

A. Profiling Research	**B. Problem Definition and Context**
• Publication characteristics (journal-conference; year; publishing outlets) • Type of paper (research paper, survey, discussion) • Study type: Does it include: 1) conceptual design, 2) implementation, 3) scenarios for experimentation, 4) case study/real-world case?	• The motivation for research (e.g., faster execution time, information sharing/hiding) • SC Sector (e.g., manufacturing, health) • SC Echelons: The number of SC actors identified from the problem context: What are the SC echelons (e.g., supplier, wafer production line, testing line)? • Clearly defined real-world problem and stakeholder involvement
C. Model Development & DS Implementation	**D. Study Outcome**
• Simulation technique (e.g., DES, ABS, SD) • Tools/Languages used (e.g., programming languages such as C++ and JAVA, COTS simulation packages like Arena^{TM} and Flexim^{TM}, computing architectures like CORBA) • Choice of middleware (e.g., HLA RTI, bespoke DS middleware like GRIDS) • Categorization of DSCS studies based on prominent research themes • Distributed simulation test beds	• Contribution: what was the contribution of the study? • Was it a generalizable contribution (Yes/No) • Proposed implementation (worked out solution given) • Real-world solution implementation and by whom • Future research: does the study indicate future research direction

adopted this second approach. Thus, a preliminary dataset was first identified using the *Web of Science*® and *Scopus* citation and journal databases, and the *ACM Digital Library* and *IEEE Explorer*. Articles were searched with the keywords *"distributed", "simulation", "supply",* and *"chain"* in the title, abstract or keywords of the publication. We included articles in the English language from 1970 until 2020 (March). Approximately 400 papers were identified through our search strategy. For every paper in the preliminary dataset, two authors reviewed the abstract. Where an abstract did not provide sufficient information to inform the inclusion decision, the authors relied on full-text reading. Articles were either included or excluded based on their relevance to the scope of the literature review. For example, several papers that dealt with distributed simulation as an enabling technology and only briefly mentioning supply chains in passing were excluded. Editorials and articles that only referred to the supply chain application as a reference example were also excluded. However, experimental results that used a supply chain application to validate a method were accepted as were panel discussions on the topic. Next, for papers included in the review, the snowballing technique was used to identify additional articles from the reference list of the aforementioned papers. At the end of the paper selection stage, we had a total of 131 articles and which constituted the underlying dataset for this review article.

Next, we developed a framework to capture and classify information from the selected papers (Table 2). We identified the following four categories of variables with each construct consisted of a number of variables that were used to describe the key aspects of the research being reviewed: (a) *Profiling Research*: variables related to the general metrics, such as article type, the source of the publication (conference, journal), and so on; (b) *Problem Definition & Context*: those that provide insights on the problem being addressed, such as the motivation for research, the application sector, supply chain echelon, and so forth; (c) *Model Development & DS Implementation*: those describing the technical characteristics of model development and implementation, such as the simulation technique, the choice of DS middleware, the experimental test bed for the execution of a DSCS, and the like; and (d) *Study Outcome:* variables that would help us identify the outcome and the contribution of a study, whether the solution was implemented and future research directions. The proposed conceptual framework **PPMO (Profiling, Problems, Models, Outcomes)** follows logical steps to give a full account of the available literature in topics that incorporate modelling techniques. The main constructs of the framework have been denoted in studies that describe methodological approaches of literature reviews and implementation science [Porter et al. 2002; Nilsen 2015]. The first construct of profiling research enhances the traditional literature review with a broad scan of contextual literature extending the span of science by better identifying and connecting efforts over research domains [Porter et al. 2002]. The elements of this construct are defined in a number of profiling studies [e.g. Mustafee et al. 2012; Mustafee, et al. 2010; Dwivedi et al. 2009]. The second construct of problem definition is fundamental in describing the situation under investigation. The elements of this construct are adapted from the conceptual framework of DES studies (stages 1 & 2) [Kotiadis et al. 2014]. The third construct covers the overarching aim of research translation into practice through models (incorporating all important aspects of distributed simulation models) and the fourth construct covers the understanding of implementation outcomes as evidence-based practice and future work [Nilsen et al. 2015]. Therefore, the methodological approach of this literature review paper is a combination of theoretical types of conceptual frameworks, such as process models [e.g., Sommerville 1996; Penedo 1989], evaluation frameworks [e.g. Proctor et al. 2011] and specific literature review frameworks [Porter et al. 2002; Kotiadis et al. 2014] in order to encompass all interesting characteristics of synthesizing research in DSCS. For capturing information on the categories, the variables were divided among two authors. The authors collected the data individually by reading the full-text articles, and the other author cross-checked the results. Figure 2 presents our literature analysis framework and Table 2 focus on the constructs' variables of the conceptual framework.

4 FINDINGS

In this section, we present the findings for each of the four categories outlined in our literature analysis that is based on the PPMO framework (Table 2). Section 4.1 is devoted to the descriptive analysis of the papers, for example, the publication outlets, findings related to the type of papers (literature review, original research articles), and implementation or case studies with empirical data. Section 4.2 is specific to the domain of application, and our analysis includes the motivation of using DSCS for SCM and stakeholder involvement. Section 4.3 is on the technical aspects of the DSCS. We have analysed papers based on model development and implementation, DS middleware, programming languages, simulation software and experimental testbeds. The study outcomes and future research directions are covered in Section 4.4.

4.1 Profiling Research (Category A)

4.1.1 Publication Characteristics. The majority of the articles were published in conference proceedings (75 conference papers- 57% of the article pool). It is not surprising since research in computer science and applied computing is frequently published in leading ACM and IEEE conferences. The dataset also consisted of 45 journal articles (34%), two books and nine book sections (9% altogether). The first paper in the dataset was published in 1997, and more than 85% of the papers were published from 2002 onwards with a peak between 2002 and 2006. Considering that distributed and parallel simulation has its origins in the 1970s and 1980s [Fujimoto 2016], this shows the comparatively late application of the technology in the context of the supply chain. However, this is not very surprising, as some of the early work was on the development of algorithms, standards and protocols for distributed simulation, and the majority of applications focused on defense and simulation of large-scale computer networks. Sixty-eight publishing

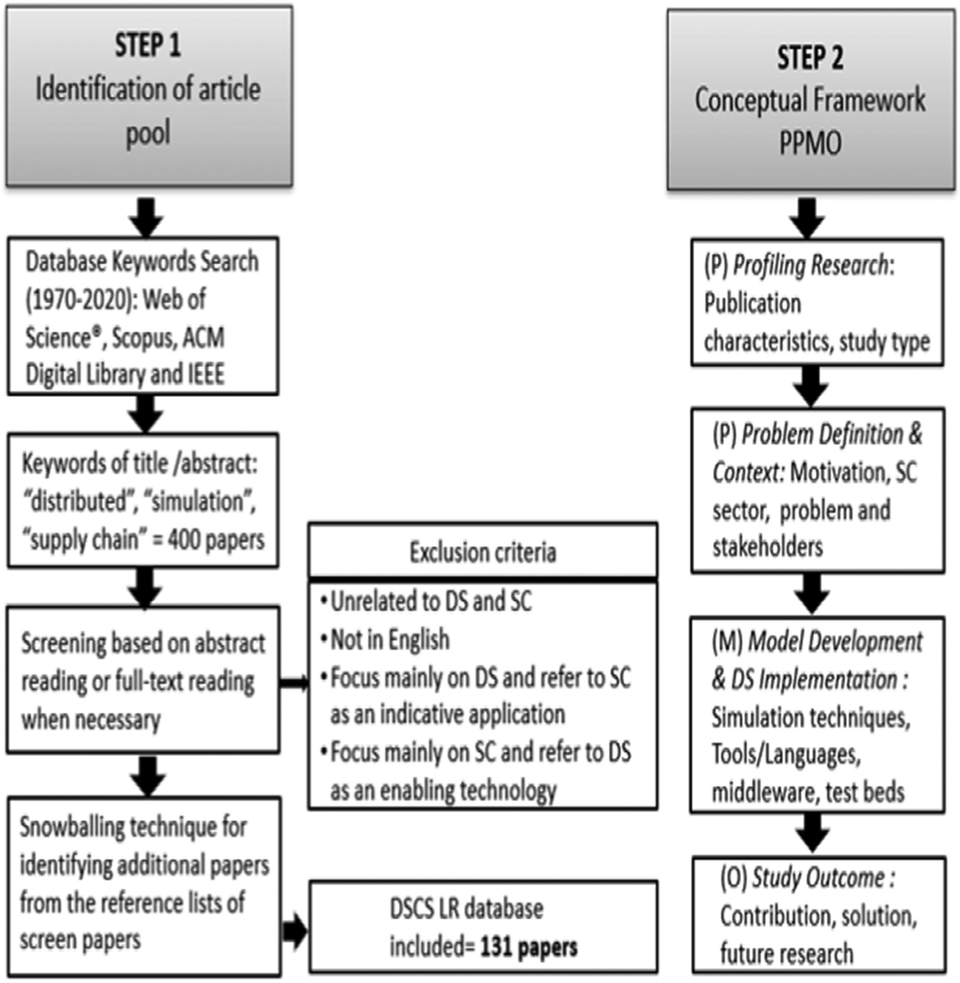

Fig. 2. Research methodology approach.

outlets disseminated DSCS research, with the top three sources being the IEEE/ACM *Winter Simulation Conference* (41 papers; 31% of all papers published in DSCS), *International Journal of Production Research* (six papers) and the *Simulation: Transactions of the Society for Modeling and Simulation International* (five articles). The remainder of the articles appear in simulation, manufacturing and **operations research (OR)**-related publication outlets.

4.1.2 Type of Paper. A first categorization is provided based on the paper's context, which makes the article, (1) a research paper, (2) a review, (3) a survey, or (4) a discussion paper. Research papers may develop a novel idea, extend methods and frameworks, or present software/model improvement. They may implement models that use generic data or data from a real-world case study, and through this, provide validity and applicability of the stated research. A research paper may hold one or more of the above elements. Review papers methodologically approach literature, and reports findings or categorize main themes. Survey papers give an overview of a topic and some future directions from a representative sample of existing work, case studies, discussion on standards, and so on. Discussion papers present the ideas of experts in the field. In our dataset, we have identified articles on multi-agent systems in SC [Lee and Kim 2008; Moyaux et al. 2006], the use of simulation in SC [Terzi and Cavalieri 2004; Mustafee et al. 2014] and DSCS frameworks [Bandinelli et al. 2006]. We have also identified surveys which utilize case studies to motivate the need for DS in real-world manufacturing and logistics problems [Lendermann 2006] and identifying interoperability issues with COTS simulation packages [Taylor et al. 2009b; Taylor et al. 2012b; Mustafee et al. 2012]. Finally, we identified discussion papers which focus on the use of DS in the industry [Lendermann et al. 2007; Taylor et al. 2002a] and on the possibility of COTS DS [Taylor et al. 2003]. However, the vast majority of our dataset consists of research papers (118 articles; 90% of all papers in our dataset).

4.1.3 Study Type. For articles that reported the development of a computer model, our literature review sought to identify whether the studies included a discussion on the various stages of an M&S study, for example, conceptual modelling, model coding/implementation, input data analysis and experimentation. In the case of input data, the distributions used in the models could be derived from either generic data or real-world case studies. A similar 'life-cycle approach' was followed in a recent review paper on hybrid simulation [Brailsford et al. 2019]. However, our life-cycle approach is specific to DS. Thus, when we refer to conceptual modelling, we do not strictly interrogate the simulation model itself and check whether the authors followed the stages of conceptual modelling [Robinson 2008], but the focus is on the conceptualization of the DS design. Similarly, for model implementation, our focus is not on the implementation of the computer model itself, but rather the DS solution and which may require interfacing a COTS simulation package with DS middleware.

- **Conceptual DS Design**: The study contributes to the conceptual design of a DSCS system (e.g., a system design that focuses on the computing and communication architecture; a feature of an algorithm to make the overall simulation run more effectively; an architectural component required for DSCS, such as the interface between a COTS simulation package, and a DS technology).
- **DS Implementation**: The study contributes to the implementation of a DSCS (e.g., the architecture design has been realized through physical hardware and bespoke software; the DS algorithms have been implemented in a programming language; a new DS middleware has been reported; the application of existing DS standards and middleware to a new case study).
- **DS Experimentation**: The study reports on relevant DS experimentation (e.g. the use of DSCS to perform "what if" experimentation on a real-world or representative case study).
- **Generic data**: The study utilizes data representative of a real-world case study (e.g. data used to test a principle in the absence of a real-world case study).
- **Case study/Real data**: The study presents a real-world application of DSCS and reports on the results.

Table 3 presents the classification of the papers based on the intrinsic characteristics of the study as per the categories defined above. We observe that although the conceptual design and the implementation stage are usually present in these papers, there is a lack of case studies and real-world scenarios. The 31 identified case studies (24%) are not necessarily unique, meaning that the same case study might have been used in different papers to examine the same or different distributed simulation issue (refer to Section 3 for a discussion). Over 55% of the articles report on experimentation; most of these experiments are conducted using generic/representative data. Twenty-three papers describe a real supply chain case study and the implementation of distributed simulation to assist decision-making.

Table 3. Percentage of Research Papers that Incorporate Specific Study Types

Study type	**% papers**
Conceptual DS design	76%
DS Implementation	72%
DS Experimentation	57%
Generic data	49%
Case study/real scenario	24%

All but ten papers are not directly related to DSCS but present technologies that assist DSCS implementation. These papers address issues that are peripheral to DSCS such as distributed databases, distributed information sharing, supply chain simulation ontologies, symbiotic simulation, cloud computing, and so on. The decision to include them in our dataset was made on the basis that these papers specifically refer to DSCS and how the research presented could contribute to the field. For example, the article by Owusu and Hauan [2006] investigates distributed information sharing; although the main body of the paper is not on DSCS and is more generic, the authors specifically note that the outcomes of the research may be relevant to DSCS. This paper is therefore included in our dataset of the ten ancillary papers.

4.2 Problem Definition and Context (Category B)

4.2.1 Motivation of Research. The motivation of the research variable examines the motivation of the study, as described in the article. Most of the papers report the anticipation of improvements in the SC performance measures and point out the advantages of using DS in supply chains. The motivation for using DS is presented below; these are listed in ascending order based on the frequency of occurrence.

- **Faster simulation execution:** Decreasing the execution time of large and complex SCS by executing different echelons of the overarching model in multiple computers.
- **Interoperability:** Functionality enhancement among multiple, disparate, heterogeneous simulation models by connecting these models into a single simulation federation maintained by DS.
- **Data hiding and sharing:** Individual models in the DSCS are linked together, and information sharing is enabled through communication between the models (e.g. each model remains in its host company and is linked to the other models as a "black box").
- **Reusability:** Reusing the individual models of each SC player in the same DSCS when the configuration of the supply chain is modified over time.
- **Geographically distributed (Web-enabling) architecture:** Inter-organizational communication is achieved by handling the execution of geographically distributed models through a single point of access.
- **Short model development lead time (this leads to lower development costs):** Simulation models can be developed by different development teams, using different simulation environments according to their expertise. As a result, the development time is reduced, and the development itself becomes easier. It could be argued that it is far more economical to link existing simulations together to form new models than to create a new supply chain model. Moreover, the ability to do this from geographically disperse places alleviates the cost and time that is usually associated with bringing participants to one physical location for conducting a joint simulation exercise.

Table 4. Percentage of Papers Referring to Different DSCS Sectors

SC sector	**Papers**
Manufacturing SC (general)	40%
Semiconductor SC	9%
Logistics and Transportation SC	8%
Automobile SC	8%
Healthcare SC	6%
SC in Aerospace; Business; Computers; Energy	6%
SC in Farm; Food; Chemical; Petroleum; Railways; Robotics; Textile and clothing sector; Computer assembly & aircraft sizing; Contracts; Electronic Commerce; Refrigerator compressors; Electric Motors; Assembly; Scheduling; Bicycle; Education	15%

- **Simulation software independence**: Linking together models created in different languages/COTS simulation software encourages simulation software independence (i.e. not all models need to be developed in the same package).
- **Large-scale/complex models coordination**: Facilitating the analysis of large systems consisting of many components interacting in complex ways such as supply chains**.**
- **Model maintainability:** Modifications to one organization unit (e.g. new plant layout) only requires one model to be changed, not the whole DSCS model.
- **Modularity:** The model of each organizational unit is usable in multiple DSCS.
- **Broader user participation** and **co-operative model development**: Executing simulations over a set of geographically distributed computers achieve broader user participation in terms of involvement in simulation experiments. This also leads to co-operative model development among different development teams and sharing of resources.
- **Scalability:** The execution times for larger simulations are comparatively more due to processing needs. DSCS allows this processing to be distributed over multiple computers, thus serving as an enabler for large SC simulations.
- **Modelling coordination:** Enabling inter-organizational SC by enhancing the process of managing dependencies between activities in SC. This helps to model the SC phenomenon called the bullwhip effect and provides the opportunity to devise strategies to counter its effects.
- **Accuracy of outputs**: Manufacturing simulations are usually "low-level" operational models. They represent system detail, dynamic behaviour, bottlenecks, resource competition, and the like. The accuracy of the models can be validated using detailed data from the manufacturing processes. DSCS enables low-level models to be linked together to enable low-level SC simulation.

The first two motives, namely, faster execution and interoperability, are the ones most frequently reported in the literature. Arguably, these two underpin the rationale of the other motivations (e.g. scalability can be achieved through faster execution and interoperability).

4.2.2 Supply Chain Sectors. Table 4 lists the SC sectors identified in our dataset. The majority of the papers make references to the manufacturing SC in general terms (this is true for 40% of the articles), meaning that they do not refer to a specific real-world case study. The scenario presented usually includes a representative set of manufacturing plants, retailers, and so on to demonstrate the use of DSCS. Nevertheless, we see in Table 3 that approximately 50% of the papers use generic data for the purposes of experimentation. The majority of the manufacturing SC papers are without a case study/real scenario tag. Twelve papers address problems in the semiconductor industry (9%).

Eleven papers on logistics (8%) deal with aspects of handling logistics network and transportation issues. The Automobile SC closely follows with ten papers, of which 40% are real case studies. Eight papers (6%) refer to SC in the healthcare sector. Very few articles consider multiple sectors. Overall, Table 4 shows that DSCS studies have primarily focused on manufacturing supply chains (general and applied). Further, although several sectors have been identified (ranging from farm and food SC to e-Commerce SC), the number of studies devoted to such sectors have been low (between one to three studies).

Examples of studies that specifically target a supply chain sector include: distributed simulation of semiconductor supply chain comprising of multiple wafer fabrication plants and assembly and test facilities [Turner et al. 2000; Lendermann et al. 2003; Chong et al. 2004], the distributed supply chain of lubricant additive industry [Fanchao et al. 2009], distributed simulation cloning using the example of a simulated supply chain comprises an agent company, a factory, and a transportation company [Dan et al. 2008], distributed simulation of an automotive supply chain [Sudra et al. 2000; Tan et al. 2003], and a healthcare supply chain [Katsaliaki et al. 2009; Mustafee et al. 2009; Anagnostou & Taylor 2017].

4.2.3 Supply Chain Echelons. The echelons are the number of stages in the supply chain. For example, a four echelon SC may consist of suppliers which provide raw materials to the manufacturers (tier-1), a production/assembly facility (manufacturer), distribution facilities, and retail operations. Our findings show that DSCS described in our dataset have modelled between two to seven echelons, with most papers reporting between two to four echelons. A good number of the papers refer to SC with generic data and therefore provide non-specific echelons. These include supplier and customers (two echelons), representative echelons of specific companies (say, company A and company B), echelons associated with the supplier, manufacturer, distributor, retailer, and consumers (five echelons). Examples of echelons from case studies/real-world include: blood supply chain (echelons for the central authority for blood collection and for hospital processes) and semiconductor manufacturing (seven echelons for wafer fabrication, wafer sorting, assembly, testing, fulfilment, warehouse, and customers).

4.2.4 Stakeholder Involvement in Simulation Study. This variable captures data on DSCS research that involved supply chain stakeholders. In our study, 19% of the papers defined a real-world problem and 8% acknowledged a stakeholder. Nevertheless, DSCS covers a wide range of research from analyzing specific SC problems to developing the underlying technology that supports DSCS. In our dataset, several papers are classified under the latter group as their research findings contribute to generalizable DSCS methods. Thus, it is encouraging to note that these papers support significant groundwork towards a general approach. From a less optimistic perspective, however, with the relatively low level of engagement with real-world problem/stakeholder, one could argue that much of DSCS research is far from being used in the industry. However, in most disciplines, research is expected to tackle equally theoretical and practical problems. In a review of M&S papers published in top OR/MS and manufacturing journals, Taylor et al. [2009a] noted that only 10% of studies supported a real-world problem and were initiated by a stakeholder. This indicates that although in the case of DSCS studies, 19% seems a low figure, it is nonetheless slightly better in comparison to the broader survey. This also holds true for stakeholder involvement. Therefore, it is arguable that DSCS research has been applied at a similar level to the broader simulation research.

4.3 Model Development and Distributed Simulation Implementation (Category C)

4.3.1 Research Themes. We undertook full-text reading of the papers to arrive at the overarching thematic structure and then classified the articles under seven DSCS research themes. This was a qualitative process and based on our reading of the literature. The themes are listed in Table 5, and the description of the themes includes up to five references.

4.3.2 Simulation Techniques and M&S Software. The majority of M&S studies in DSCS have applied the following three modelling techniques - DES, ABS and SD. DES and ABS are examples of discrete-time simulation, and SD is continuous-time. Discrete-time is considered the modelling paradigm of choice since the majority of the studies have the underlying scope of investigating the short-time behavior of supply chains. Although ABS has been applied widely in the context of social simulations - refer to Li et al. [2008], who present a review of agent-based social simulation – its application to supply chain is comparatively recent. Moreover, we did not find any studies on microsimulation, which is yet another technique predominantly used in social sciences but now applied to modelling traffic and transportation. Like ABS, microsimulation also models entities at an individual level. However, the former enables the development of more sophisticated models as it allows modellers to incorporate agent-level decision making, inter-agent interations, etc. Q2 [Gilbert 2019], and which allow for the better representation of SCS models. It is therefore not surprising that in the discrete-event space, DES is the simulation technique which outweighs studies that have used ABS by almost 3:1 (64 papers have discussed DSCS in relation to DES, compared to only 23 studies that have reference ABS and multi-agent systems). Included in the numbers presented are six hybrid simulation studies which have used a combination of DES and ABS techniques. SD has only been used in the context of hybrid DSCS studies combining SD with DES [Venkateswaran & Son 2004; 2005; 2009]. Our literature review has also identified two distributed simulation games for SCM training, a numerical simulation that uses MATLAB and Simulink for dynamic management of supply chain networks and some examples of numerical optimization, simulation-optimization and interoperable simulation in the context of DSCS. We present our findings under DES, ABS and Hybrid M&S, respectively; we also identify the **commercial off-the-shelf (COTS)** simulation packages that have been mentioned in the papers. However, not all articles have provided specific implementation details and/or describe the model and middleware integration as future work.

DES Studies: Our analysis has shown that the commercial off-the-shelf (COTS) simulation software Arena had more than 20 references. It has been used with middleware such as MAK RTI [Gao et al. 2008], HORUS^{TM} middleware [Revetria et al. 2003], with a message-oriented middleware called **Microsoft Message Queue (MSMQ)** which was used to link simulation models [Saad et al. 2003], with Transaction Coordinator Server [Rathore et al. 2005] and also with web services technology [Lee et al. 2007]. Arena^{TM} was also discussed in the context of the NIST **Distributed Manufacturing Simulation (DMS)** adapter [Son 2005; Jain et al. 2007]. The NIST DMS adapter [McLean and Riddick 2001] was developed to provide mechanisms for distributed simulation similar to those provided by the HLA RTI, but which was specific to the manufacturing community. Simul8 was referred to in seven papers. It has been integrated only with DMSO RTI(1.3NG [Mustafee et al. 2009]. Other COTS pages that have been referred to include Automod/ASAP, AutoSched AP, Delmia QUEST, and AnyLogic. Authors have also programmed DES simulation executive using Java, C++, Simple++ and have used libraries such as the **Distributed Simulation Object Library (DSOL)**. A total of ten studies have implemented DES software using these languages and libraries.

ABS Studies: The JADE agent development framework is the most popular ABS development toolkit. **JADE** is an acronym for **Java Agent DEvelopment Framework** and has been referred to by nine studies. It has been used with pRTI middleware [Jian et al. 2017]. Authors have also used JADE to implemented HLA-compliant event-driven time synchronisation protocols for

Table 5. Prominent Research Themes

Research Themes	**Description**
DSCS Middleware Development (including time management protocols)	Articles in this theme include enterprise-wide DSCS using HLA and MPI- Asynchronous Simulation Protocol [Gan et al. 2000a], development of **Generic Runtime Infrastructure for Distributed Simulation (GRIDS)** [Sudra et al. 2000; Tan et al. 2003] and the GRIDS-**SCF (Supply Chain Federation)** middleware for distributed supply chain simulation [Taylor et al 2002b], development of **Distributed Manufacturing Simulation (DMS)** adapter to integrate legacy simulation systems with HLA runtime infrastructure [McLean and Riddick 2001], development of a **Java-based Distributed Simulation Object Library (D-SOL)** to simulate a real-world supply chain [Jacobs et al. 2002], development of new time synchronization approach for distributed simulation (Epoch Time Synchronization) and benchmarking using a manufacturing supply chain simulation [Rathore et al. 2005]; development of a hybrid conservative approach for time synchronization called **Advanced Look-ahead Based Approach (ALBA)** and its application to a distributed supply chain simulation comprising of assembly shop, supplier, body shop, and drive train shop [Tammineni and Venkateswaran 2007].
Reference Models for DS in the Industry (including SCM)	Our review identified two prominent reference models for use with DSCS. **(a)** The *HLA-CSPIF Type I to VI Reference Models* that were developed for the purposes of communicating concepts and problems between researchers, users, and vendors [Taylor et al. 2012b]. **HLA-CSPIF** is an acronym for The **High Level Architecture-COTS Simulation Package Interoperability Forum**, and whose purpose was to create a standardized approach to distributed simulation using the IEEE 1516-2000 High Level Architecture to support the interoperation of discrete event models created in Commercial Off-the-shelf simulation packages. This work ultimately led to the development of the **Simulation Interoperability Standards Organization (SISO)** Standard for Commercial-off-the-shelf Simulation Package Interoperability Reference Models (*SISO-STD-006-2010*) [SISO 2010]. The reference models were used for the development of the National Blood Service distributed supply chain model [Mustafee et al. 2009]. **(b)** *Business Process Reference Model*, developed as part of the MISSION project [Rabe and Jaekel 2003], integrated approaches such as SCOR, **Integrated Enterprise Modelling (IEM)** and distributed simulation methods into a consistent reference model for distributed supply chain design and configuration [Rabe et al. 2006]. Jain et al. [2007] reported the use of an existing specification called **OAGIS (Open Applications Group's Integration Specification)** for interoperability testing among partner organizations in a supply chain. The work was sponsored by the **National Institute of Standards and Technology (NIST)** Manufacturing Interoperability Program. The program supported projects applying information technologies and standards-based approaches to manufacturing software integration problems. The funding of this work by NIST shows the importance of DSCS in the overall context of interoperability of manufacturing and supply chain systems.
Extending ***Supply Chain Operations Reference (SCOR)*** *Model for DSCS*	The SCOR model, established by the Supply Chain Council, now called the Association for Supply Chain Management (http://www.apics.org; last accessed March 2020), provides a reference model for analysing a supply chain that is based on a standard set of variables and metrics. Based on SCOR, Barnett and Miller [2000] implemented the e-SCOR modelling and simulation environment based on a discrete-event simulator and the HLA standard to simulate a virtual enterprise. Fayez et al. [2005] developed an SCS ontology, which is based on the SCOR model that could be used to define supply chains and to capture distributed knowledge to build DSCS.
Methodological Aspects of DSCS	Articles in this theme include the use of DSCS to enforce the confidentiality of the data provided by supply chain network partners [Mekaouche et al. 2009], fault-tolerant distributed simulation to decoupled federate architecture – also referred to as distributed simulation cloning [Dan et al. 2008], information hiding in the supply chain [Cai et al. 2001], application of DSCS in the context of inter-enterprise planning to enable protection of intellectual property rights and keeping enterprise models private [Mertins and Rabe 2005; Mertins et al. 2005], DSCS with integrated Advanced Planning and Scheduling procedures for collaborative supply chain optimization [Lendermann et al. 2001], and a distributed simulation-based approach for supply chain interoperability testing [Jain et al. 2007].
Discrete Event System Specification (DEVS) for DSCS	Examples of papers in this theme include the development of **Generalized Discrete Event Specification (G-DEVS)** models and **HLA (High Level Architecture)** standard for RFID-based freight transportation system [Zacharewicz et al. 2011], development of a DEVS Class Hierarchy for SCM implementation and the development of prototype design of DSCS using DEVS/CORBA run time infrastructure [Zeigler et al. 1999], modelling autonomous control entity in the context of a holonic manufacturing system where the automated entities are modelled according to the DEVS formalism and integrated to an HLA-enabled DSCS [Mekaouche et al. 2009], and a DEVS-enabled simulator architecture that is implemented using the ***Complex Adaptive Supply Networks Simulator (CAS-SIM)***, which is a distributed agent-based tool for DSCS [Pathak et al. 2004]. CAM-SIM is used to model the growth dynamics of automobile supply network in the US.
Hybrid DSCS	Examples include the work by Venkateswaran and Son on the use of HLA for coordinated execution of hybrid simulation models developed using Discrete Event and System Dynamics [Venkateswaran and Son 2004; 2005; 2009], distributed simulation for modelling of offshore wind farms using ABS and DES [Mustafee et al. 2015], and a hybrid ABS-DES simulation of healthcare supply chain comprising of an ambulance network that serves several hospitals [Anagnostou and Taylor 2017].
Internet-driven DSCS	Examples include DSCS using web services and BEML [Low and Turner 2006], symbiotic simulation for real-time decision making in the context of the supply chain of lubricant additive industry [Fanchao et al. 2009], and developing an architecture for internet-mediated, service-based distributed simulation games and its implementation as a supply chain simulation game [van Houten and Jacobs 2004].

multi-agent distributed simulation [Long et al. 2011]. Repast has been referred to by two studies. Yoo et al. [2009] combined Repast with JADE and used a JADE sub-class to track global simulation time and to enable synchronization, and Anagnostou and Taylor [2017] used Repast with Portico v2.0 RTI middleware. Studies have also referred to agent toolkits/languages such as Zeus Agent Building Toolkit, Agent Unified Modelling Language and Jason agent reasoning framework.

Hybrid M&S Studies: SD is a modelling approach that has applications in operations strategy as it relies on a holistic approach for system investigation. SD is identified in only four DSCS studies and as a *hybrid method* together with DES. This is primarily the work of Venkateswaran et al. [2005] and Venkateswaran & Son [2005, 2009] and they have used multi-resolution hybrid modelling with SD and DES to analyse the local planning decisions of a three-echelon supply chain and its impact on enterprise-level planning. For these studies, PowerSim and Arena were used as SD and DES COTS packages respectively. We have identified two papers that have used ABS with DES for DSCS. The work by Anagnostou and Taylor [2017] present a distributed simulation framework for OR/MS applications and have used Repast Harmony with Portico v2.0 RTI implementation to model the ambulance supply chain.

4.3.3 Distributed Simulation Standard and Middleware. A simulation has to process events in increasing timestamp order. Failure to do so will result in causality errors. Synchronization protocols are used to prevent causality errors from occurring. They can be broadly divided into conservative synchronization protocols and optimistic synchronization protocols. In a conservative protocol, a processor is never allowed to process an event out of order; whereas in an optimistic protocol, a processor is allowed to process an event out of order, provided it can revert back to its previous state in the case of a causality error [Nicol and Heidelberger 1996]. A distributed simulation middleware is a software element that applies the conservative and optimistic algorithms to attain synchronization among the separate running simulations. Examples include **Aggregate Level Simulation Protocol (ALSP)** [Wilson and Weatherly 1994], IEEE 1278-1993 **Distributed Interactive Simulation (DIS)** [Miller and Thorpe 1995], GRIDS [Taylor et al. 2002b], CSPE-CMB [Mustafee, 2004], FAMAS [Boer, 2005], and IEEE 1516 **High Level Architecture – Run Time Infrastructure (HLA-RTI)** (IEEE 2010]. Our study does not differentiate among the alternative distributed simulation frameworks that are employed for modelling the SCs. Nevertheless, we wish to make some important observations: (a) Distributed simulation middleware such as DIS and ALSP have been extensively utilized in simulations of defense training, but no use in civilian applications has been reported; (b) The HLA, while it initially has been developed to address the requirement of interoperability among existing and new simulations of the U.S Department of Defense, since then it has been largely recognized as the de-facto standard for distributed simulation, now as an IEEE standard. In many cases the HLA standard together with its middleware (HLA-RTI) have been used for distributed simulation applications in the civilian sector; (c) Numerous other middleware have been created in universities aiming at the facilitation of distributed simulation in the business world, for example, GRIDS [Taylor et al. 2002b], FAMAS [Boer 2005], CSPE-SMB [Gan et al. 2000b; Gan and Turner 2000] and CSPE-CMB [Mustafee 2004] – both of which are an extension of Chandy, Misra, and Bryant distributed discrete-event simulation algorithm [Chandy and Misra 1981; Bryant 1977]. However, these are mostly software developed for a specific project apart from the **Service-Oriented HLA-RTI** [Pan et al. 2007], **SOHR** for short, created by the **Parallel and Distributed Computing Centre (PDCC)**, Nanyang Technological University, Singapore. Table 6 lists the distributed simulation standards and middleware discussed in the papers. Not all studies that mention the middleware present an implementation. Refer to section 4.3.5 for a discussion on implementation testbeds.

Table 6. Papers that refer to Distributed Simulation Standard and Middleware

Middleware/Standard	Instances	Comments
HLA-RTI	38	Under 'HLA-RTI', we group studies that refer to both the HLA standard and the HLA-RTI. The RTIs that have been reported include DMSO RTI, RTI+ extended RTI Middleware [Gan et al. 2003], MAK RTI, poRTIco, Pitch pRTI.
HLA and WS-HLA	18	Under 'HLA and WS-HLA' we list studies that mention the HLA standard, without explicitly referring to the RTI. For example, articles that mention the use of HLA-based Distributed Simulation Manufacturing Adapter (3 instances), use of **Web Services** technologies with the **HLA** - **WS-HLA** [Hongyu 2010], HLA-complaint time synchronisation mechanism integrated with a bespoke solution [Long et al. 2011; Wartha et al. 2002]. In work reported by Wartha et al. [2002], the **DST-SC** tool **(Decision Support Tool-Supply Chain)** is an extension of AnyLogic^{TM}.
Bespoke Solutions	21	'Bespoke solutions' are DSCS that have implemented bespoke middleware and which does not adhere to the HLA standard. Examples include the implementation of a middleware architecture that supports rollback [Klobut et al. 2007], bespoke SYNCHRO middleware [Lannone et al. 2007], implementation of a message-oriented middleware [Saad et al. 2003], Distributed Simulation Object Library [Jacobs et al. 2002; Jacobs 2005; van Krevelen et al. 2011], GRIDS [Taylor et al. 2002b], Transaction Coordinator Server [Rathore et al. 2005], and CMB-SMP [Gan et al. 2000b; Gan and Turner 2000]. Five papers refer to the GRIDS middleware; Transaction Coordinator Server and CMB-SMP have been referenced in two publications respectively.

4.3.4 Integration Approaches. This section is a discussion on integration approaches that are commonly used for linking COTS packages and bespoke simulation code with distributed simulation middleware. None of the existing COTS simulation packages lends themselves readily to distributed simulation, and the integration of these packages with the middleware is essentially done through bespoke solutions which require coding. For example, the use of bespoke VBA code [e.g., Bandinelli and Orsoni 2005] and Java [McGinnis 2004] for information exchange between the package and the middleware. Our analysis shows that databases and spreadsheets have been used as an intermediate agent for communication between models. In addition to Excel/VBA, authors have referred to MS SQL 2000 database (three instances), MS-Access, SQL and Oracle DB (one instance each). Authors have also referred to the distributed network database and supply chain network database. As mentioned earlier, researchers have also developed simulators by coding them in programming languages such as C/C++ and Java, and in this case, the integration approach is seamless since the same programming language can interface with the middleware. For example, RTI 1.3NG has both Java and C++ bindings. We identified 14 papers that mentioned the use of either Microsoft VBA, Visual Basic or the .NET framework as the programming language for interfacing COTS packages with middleware. For example, Saad et al. [2003] and Linn et al. [2002] used Arena^{TM} COTS package with VBA; Lee et al. [2007] used Arena^{TM} and Visual Basic with Web Services technology (instead of HLA/RTI) to integrate heterogeneous simulations. Twelve studies mentioned the use of Java, of which's five studies also mentioned the use of *Java Native Interface* (this is essential if the COTS package is not developed in C/C++ and Java bindings to the HLA are used). McGinnis [2004] used Java with COTS package Automod^{TM}, Jeong et al. [2009] used Java Server Pages with a database for the development of a web-based simulator and Sudra et al. [2000] used Java with the GRIDS middleware. Examples of C++ as an integrated approach include the work of Gao et al. [2008] who used C++ with Arena^{TM} and MAK/RTI, Turner et al. [2000] who use

Table 7. Papers that refer to Distributed Simulation Standard and Middleware

Number of computers in the experimental test bed	**#Num studies**
Two computers (Test bed = 2)	6
Greater than two and less than or equal to five computers (2 < Test bed 5)	20
Greater than five and less than or equal to ten computers (5 < Test bed 10)	6
Greater than ten and less than or equal to 15 computers (10 < Test bed 15)	1
Local (one-computer) experimentation or vague about the underlying configuration of the test bed	25

C++/Active Threads and Teixeira et al. [2004] who used Visual C++ to develop an Internet-based distributed application for supply chain management training. Some studies have relied on more than one programming language to achieve this integration. For example, Venkateswaran et al. [2005] used COTS packages Powersim$^{\text{TM}}$ and Arena$^{\text{TM}}$ with VBA and C++, Tammineni and Venkateswaran [2007]'s work on look-ahead based approach for DSCS was implemented using Java, C++ and Arena$^{\text{TM}}$, and Mustafee et al. [2009] used Java/JNI and VBA with Simul8$^{\text{TM}}$.

4.3.5 Distributed Simulation (DS) Test Bed. In this section, we examine the test beds that were used for simulation experimentation. We refer to the test bed as the number of inter-connected computers (LAN, WAN, Internet) that were used for executing a DSCS. A total of 58 papers included a reference to experiments. As can be seen from Table 7, 25 studies were vague in relation to the configuration of computers used for the experiments. The majority of the studies executed distributed simulations over test beds that comprised of more than two and less than or equal to five computers (20 studies).

Our findings suggest that several studies include one computer implementation of DS. In this case, multiple federates are executed in the same computer as separate processes and/or over a computer with multiple cores; such studies usually discuss the applicability of multi-computer distributed execution in very general terms. For example, Barbuceanu et al. [1997] present an agent-based design of a supply chain using a **coordination language (COOL)**. In this work, the authors execute the simulations locally and only refer to the fact that the agent language supports distributed execution of agents using TCP/IP. Similarly, although Barnett and Miller (2000) developed a bridge between e-SCOR (a process model development environment based on SCOR and layered upon a DES) and the HLA, they have been vague about whether the DS was implemented ("Any component of the model can be distributed amongst a federation of machines using the infrastructure"). Other papers have implemented distributed simulation but have not provided specific details on the experimental test bed. For example, Nurmilaakso (2004) implemented a prototype of a supply chain scheduling system based on parallel forward simulation and agent architecture over the Internet, but no further details are included. Similarly, Xu and Lin [2009] implemented a new time advancing mechanism for agent-oriented SCS using JADE and the HLA, but in terms of the test bed there is no information available. Another example is the MAK/RTI implementation of a DES distributed simulation for container terminals using Arena, C++ and a Database software [Gao et al. 2008], and although the authors mention four federates they do not provide explicit information on the DS test bed. Finally, there are several papers in our data set which have specifically mentioned implementation as future work. For example, the implementation of a Web Services-based HLA distributed simulation method (WS-HLA) for the analysis of bullwhip effect and information sharing in the supply chain [Hongyu 2010] and the implementation of prototype design of DSCS based on DEVS/CORBA run time infrastructure [Zeigler et al. 1999].

In the remainder of this section, we present examples from papers that specifically report on DS test beds. We identify the tools and simulation packages used, the choice of DS middleware and underlying configuration for experimentation. Yoke et al. [2006] experimented with an implementation of HLA-based DSCS which included a test bed of three computers which hosted the web portal server, the **Business Process Execution Language (BPEL)** and DMSO RTI. An experimental set-up of five computers was used by Mustafee et al. [2009] to execute a blood supply chain simulation, wherein four computers were running a copy of the CSP Simul8^{TM} and the fifth computer executing the RTI process. Bruzzone et al. [2002] experimented with a HLA federation that comprised of 7-8 federates, each of which executed simulation models developed in either Java or CSPs like Arena and Simple++. The authors tested 30 simulation runs of the federation (15 in Wide Area Network and 15 in Local Area Networks). Cai et al. [2001] used RTI 1.3NG-v2 with a discrete event simulator developed in C++ and executed this over a test bed that included five workstations. Dan et al. [2008] study on developing a fault-tolerant HLA-based distributed simulation by creating virtual federates (also referred to as cloning) relied on a test bed comprising of between 3-12 workstations and one server computer. Experimentation related to a new hybrid distributed simulation based on Parallel Federate protocol [Ji et al. 2001] relied on one 4-CPU shared memory system, three workstations, and one server computer with RTI executive and federation executive processes. Lendermann et al. [2003] presented an HLA-based supply chain simulation for the semiconductor industry (with federated developed in both C++ and Java) and the DS was executed over both LAN (comprising two multiprocessor machines and a workstation) and WAN. In the case of WAN, the models were executed in two sites in Singapore and one in Oxford University. In Rathore et al. [2005], the authors implemented a transaction control server which acted as a middleware for distributed execution of DES models developed in Arena^{TM}. In this work, the simulation experiments were spread across five computers, one for the transaction coordinator and one for each of the four federates.

4.4 Study Outcomes (Category D)

4.4.1 Contribution of the Articles. The contribution of the studies entailed in our literature review provides further insights which add upon the knowledge gathered at the "Motivation" variable in section 4.2.1. The contribution is split into two categories: (a) Specific contribution which is mapped to the motivation of the study and acknowledges whether the original motivations of the study were attained. In case of partial attainment of the original objective of the study, the authors might indicate future work for the accomplishment of the remaining objectives; (b) Generalizable contribution are findings that could be used by other studies, for example, a framework, an architecture, a standard, or an algorithm. Our findings show that along with the review and discussion papers, almost all the research studies contribute to the motives of the study as presented in their abstract or introduction and as presented in the motivation of the research section of the paper. However, only 53% of the articles report generalizable contribution (Table 8), for example, model synchronization mechanisms and algorithms for time advance.

4.4.2 Problem Solution and Implementation. The next two variables report on the usefulness of the proposed DSCS solution and whether implementation was mentioned explicitly in the published paper. The first variable reports on whether the proposed solution solved the real-world stakeholder problem. The second variable sought to evidence whether the solution was implemented in reality and reported the details provided on the implementation conditions and setting. From our database, it is noted that only 8% of the papers (10 articles) report a useful solution and one or two refer to the implementation of this solution. Studies which provide a valuable solution to the imposed problem are described in: papers which use DS to speed up the model execution time of the previous monolithic DES model (e.g. Katsaliaki et al. [2009]) or to enable the execution of a large-complex system [Rossetti and Chen 2012]; a paper which applies DS to the manufacturing

Table 8. Generalizable Contribution of the Studies

Algorithms
Algorithms for the reduction in overhead of the RTI, look-ahead algorithms for timestamp messages, time advancing mechanism, time bucket-based resource reconciliation mechanism, message-driving formalism, communication-negotiation protocol, use of statistical inference method for determining the appropriate timing of communications among federates.
Interfaces
Interface SD and DES models, COTS handler, HORUS middleware that acts as a Delegate Simulator for COTS handling, demonstrator for connection between different simulation models, development of an adapter to realize the corresponding interface between federates and library of RTI.
Development Processes (including Standards)
GRIDS Federation Development Process, object exchange model template for GRIDS, modelling coordination technology, developing federation architecture and infrastructure, use of multi-agent systems for DSCS, DSCS modeling method based on **HLA, Agent and SCOR** (also referred to as **HAS**), ontology network for DSCS, *SISO-STD-006-2010* Standard for Commercial-Off-The-Shelf Simulation Package Interoperability References Models, standardization of the interoperation of COTS simulation packages via the HLA.
Frameworks
Framework for supporting runtime robustness to HLA-based distributed simulations, a framework for advanced distributed simulation with integrated APS procedures for collaborative supply chain optimization, a framework for hybrid DS, a framework for analyzing DSCS, a framework for choosing the right system according to the motive of the DCSS, standardized validation process for DS models, reusable supply chain models for DS, a framework for reusability of legacy federate code, standardized validation process for DS models, framework linking together the methodological practices of OR/MS and DS using only the minimum required HLA functionalities.
Prototype Architectures
DS prototype for SC, a prototype of SC scheduling system, a prototype of distributed semiconductor SC simulation, a reusable client-server, web-enabler architecture, adaptable system for configuring and testing SC, growth-oriented simulation model for supply networks

and installation of railways' switch point assemblies and shows that the improved communication among simulation federates makes the scheduling tool more responsive to the dynamic needs of production and installation, increasing on-time deliveries and reducing the cumulative delays on late deliveries [Bandinelli et al. 2004]; research which refers to the initialization of a distributed logistics simulation without the need of a warm-up period [Dalal 2003] and two papers which use HLA to achieve reusability, interoperability, data hiding and evaluation of several management policies for decision-making in an extensive aerospace supply chain [Bruzzone et al. 2005], and in emergency care (ambulances-hospitals) [Anagnostou & Taylor 2017]. Taylor et al. [2009b] report on a study conducted at *John Deere & Company*, a manufacturer for agricultural, forestry, and construction equipment. The results of the DS model were implemented at the tractor production system at its South American factory. The achievement of the use of DS was to integrate different independently developed *existing* models that could not be combined within a single CSP for joint execution [Taylor et al. 2009b].

This low level of DSCS implementation in support of real-world problem solving is hardly surprising. None of the COTS simulation packages currently supports DSC. As a result, practitioners and industrial partners often collaborate with researchers to develop standards-based or bespoke

DSCS solutions. However, it is difficult to transfer the skills from the research centres to the industries since the collaboration usually takes the form of research projects, and there is no formal training or long-term support of the codebase. Furthermore, the development of distributed models often requires not only the M&S skill-set but also expert knowledge of distributed systems and applied computing. It is often difficult to find this mix among simulation practitioners, who are arguably best placed to implement the real-world DSCS studies. Therefore, even if researchers and practitioners recognize the use of DS for supply chain design and analysis, the technologies are not yet adopted and diffused at an industrial level, making commercial implementation problematic [Bandinelli et al. 2006].

4.4.3 Agenda for Future Research. A total of 96 papers have been identified to report future work, approximately 73% of the 131 articles in our dataset. We gathered this information by reading the concluding section of the papers (which frequently list future research directions) and by conducting a full-text keyword search of the word 'future' to identify sections associated with future work. The reported work was classified under two groups, (a) general/broad-ranging future work (including, for example, significant challenges, new/improved methodologies, new research direction, inter-disciplinary exploration, development of new tools/language), and (b) specific work identified by the researchers (including algorithm/architecture enhancement/extension, further research artefact implementation, additional experimentation and validation, adding more echelons to the modelled SC, extending the number of SC players included in the model, and applying the presented technique to different problems in the domain or a different field). A quarter of the studies have identified both general and specific directions for future research. Table 9 lists the future research directions under the following three themes – Application, Methodology and Technology.

5 DISCUSSION

The discussion section presents a synthesis of our analysis in the form of motivations, challenges, and risks in applying DS to supply chains (Table 10). It also presents recommendations for businesses and academia. These recommendations were earlier discussed in the results section, especially in the analysis of future work.

It is also interesting to note that the DS technology and standard that was initially developed for the military has been used in the industry, albeit with a reduced feature-set. The DS protocols, like the ***Aggregate Level Simulation Protocol (ALSP)*** and the ***Distributed Interactive Simulation (DIS)***, were developed by the military and for military applications [Dahmann et al. 1997]. Subsequently, the HLA standard was developed with the objective of providing a common architecture for supporting reuse and interoperation of simulations across the U.S. Department of Defense (*ibid.*). HLA not only became the defacto standard for ensuing interoperability of U.S./NATO military simulations but, in the context of the present study, it is perhaps more important to acknowledge *how* the standard was recognized to be of benefit to the industry. The *National Institute of Standards and Technology (NIST)* DMS adapter was developed for the manufacturing industry with the aim of providing DS mechanisms similar to those provided by the HLA RTI [McLean and Riddick 2001]. Considering that the first book on the implementation of DS was published only back in 1999–2000 [Kuhl et al. 1999; Fujimoto 2000], and the IEEE HLA standard introduced in 2000, it should come as no surprise that DS has not been fully embraced in the industry. Taking a cue from the history of DS, greater adoption of DSCS could be realized by encouraging cross-disciplinary partnerships among Operational Researchers, experts in distributed systems and applied computing, software engineers, and vendors of COTS simulation package. Such partnerships will encourage the rethink of the traditional approaches to DSCS and will inform the development of integrated and hybrid M&S solutions for future supply chains.

Table 9. Agenda for Future Research

A. APPLICATION
Broad: Increase use of DS in the supply chains; Increase use of DS in the industrial sector; Investigate the use of DS in disaster incident management and emergency response operations; Enhance collaboration with industry; Acquire more application experience; Use of a real supply chain system (instead of a simulation) with Distributed Hybrid applications; Simulate continuous manufacturing and trading; Develop a real-time-based simulation framework and real-life applications; Compare architecture with different configurations, for example with other simulation engines, reasoners, or individual actor implementations.
Specific (refers to specific studies/frameworks): Apply the MISSION technology [Rabe and Jaekel 2003] for training purposes; Extend from an enterprise framework to a general supply-chain framework; Investigate the benefits of Netcentric Management in organization change and human resource management; Compare prototype alternatives; Apply the message-driving formalism in the simulation of complex multi-agent systems; Apply the (discussed) architecture in an expanded version of the case study with more echelons; Test the robustness for the (discussed) model; Test the technologies in a real scenario; Create a library of ready-to-use simulation components; Apply (approach presented) in other industries; Better data acquirement.
B. METHODOLOGY
Broad: Use of ontologies for information exchange between supply partners; Application of agents to SCM problems; Improve agents representation of SC echelons; Application of Artificial Intelligence based on Neural Networks for estimating SC performance measures; Application of Artificial Intelligence to SCM; Create a multi-agent simulation platform dedicated to heterogeneous behavioural studies of SC; Apply learning Intelligent Agents; Compare hybrid model with hierarchical production planning; Extended decision tools in SC models; Scalable information sharing; Extend hierarchical planning system approach integration; Achieving parallelism in simulation computation; Further development of the infrastructure to other programming languages and simulation techniques; Investigate bounded searching, federative grouping, flexible boundaries, and selective pattern recognition in DS; Develop a general **parallel and distributed simulation (PADS)** architecture; Investigate other architectures that may efficiently support the agent communication approach; Use of regression meta models as a performing alternative to long simulation run time of COTS; Development of metrics for what can be distributed; Create a step-by-step guide to show how to modify existing OR/MS simulations to become part of a federated DS model.
Specific: Improve portability and flexibility of a distributed application; Develop simulation modeling for push production; Validate and enhance the components of the architecture that is proposed in a paper (several instances); Test the validity of the proposed architecture; Improve filter for increased accuracy; Improve simulation analysis; Develop specifications; Define transformation rules for mapping HLAFed concepts in a **FOM (Federation Object Model)** and rules to transform SC knowledge concepts in simulation models.
C. TECHNOLOGY
Broad: Improve interoperability; Improve COTS handler; Improve distributed capabilities in COTS; Compare distributed serial simulation with distributed parallel simulation; Model ynchronization in a distributed way instead of a ynchroniza way; Improve time synchronization between remote services and between visualization and simulation services; Compare time advancing mechanism; Investigate security aspects of the network connections; Improve security of the simulation systems within WAN; Compare external communication between the virtual and physical federates; Create algorithms for decision making; Investigating the performance of the SC with different RTIs (e.g., MAK RTI, ynchro, Pitch pRTI); Test different standards such as the **Core Manufacturing Simulation Data (CMSD)** standard under development by SISO (*SISO-STD-008-2010*); Create a commercially sound proof-of-concept solution for federated simulation; Use semantics to solve the syntactical differences between similar entities in a federation; Using an ontology language; Create web-enabling technology; Create online web service; Development of interfaces for use of web services; Improve technology standards, web services processes orchestration and choreography; Create online web database for simulation builders.

(Continued)

Table 9. Continued

Specific:
Automate the re-structuring of the business process; automate the evaluation of workflows; Automate the generation of the code for data conversion and information hiding; Create an automatic decision support system; Improve the simulator; Store the ontology in XML schema or even ACML schema; Construct the super-FOM; Improve the look-ahead algorithm for more complex systems; Improve asynchronous timing conditions; Test the scalability of the look-ahead algorithm; Test the scalability of time ynchronization algorithms; Incorporate time synchronization scheme; Improve the Fault Tolerance mechanism; Interface with partners; Interface with other programming languages; Create online database and repository; Improve the DMS adapter implementation to .Net technology; Make the PF protocol applicable to all the object management services of the RTI; Test the framework with COTS; Increase graphical capabilities of the architecture; Translate the agent ontology from the KOJAC software design steps directly to XML/ACML-based semantics; Create SCOR tasks' library for use in simulation modelling.

From the analysis presented in Table 1, it is apparent that using DS approaches to SCS has the potential to create large-scale, flexible supply chain decision support tools based on private, detailed, reusable simulations. It seems to be the "natural choice" for modeling the reality of SCs, consisting of loosely coupled and disperse sub-systems which share selected information but otherwise act independently [Anagnostou & Taylor 2017] Simulation is classified as a prescriptive analytics approach [Lepenioti et al. 2020]. A DSCS thus allows for experimentation of strategies between different tiers of the supply chains and leads to more informed decision making. Taking the example of the ongoing COVID-19 pandemic and the global vaccination effort, the supply chain for vaccines extends from research universities/pharmaceutical companies that originally developed the vaccines (e.g., Oxford-AstraZeneca, Pfizer/BioNTech, Moderna) to large-scale manufacturers (e.g., the Serum Institute of India which has gained the license from Oxford-AstraZeneca to produce at least a billion doses of the vaccine), and further downstream to the regional vaccination storage sites and eventually to the vaccination centre. Computer models of the different echelons of the supply chain may exist; for example, Pfizer/BioNTech and Moderna may have DES models of their vaccination manufacturing facilities; a logistics provider may have an existing network flow simulation model of cold transport and storage. Both Pfizer/BioNTech and Moderna produce mRNA vaccines that require ultracold storage and may be served by the same logistics provider. A distributed model comprising of vaccine production facilities (model A and B) and cold storage and transport (model C) would enable both the manufacturers and the logistics provider to experiment with strategies that will enable the latter to maximise its use of resources to allow a controlled roll-out of both vaccines, also taking into account the capacity of the numerous vaccination centres and which may have its model (model D). Thus, by linking the models, it would be possible to experiment with configurations of change in manufacturing capacity and its outcome [Currie et al. 2020] in terms of logistics providers' resource utilisation (e.g. models A-B-C); the effect of a change in the capacity for administering the vaccine, for example, the availability of nurses and volunteers at the various centres, and its outcome on the inventory of vaccines in cold store (models C-D); the effect of logistics-related disruptions and its implications for the manufacturers and unmet demand at the vaccination centres (models A-B-C-D). The development of such simulation-based decision support tools requires collaboration among organisations that are part of the vaccine delivery supply chain; it requires shared access to data and models. One example of this is the US-based ***COVID-19 Healthcare Coalition (C19HCC)*** which has galvanised healthcare organisations, technology companies and NGOs into a coalition that used several computational approaches to help decision-makers make better sense of COVID-19 data from heterogeneous data sources [Tolk et al. 2021]. Although the context here was mainly data orchestration, forecasting and visualisation of data, a collaboration such as C19HCC could help rapid development of DS for COVID-19 SCS.

Table 10. Considerations for DSCS

MOTIVATIONS: • The benefits of using DSCS as described in section 4.2.1 (e.g., faster simulation execution, interoperability, reusability). • Free access to DS middleware, for example, OpenRTI and Portico.
CHALLENGES: • Most of the recommended solutions for adopting DS environments cannot be integrated with COTS simulation packages. Where such solutions exist, for example, DSCS using models developed in COTS packages, they are generally research projects that are conducted with industry partners. This represents a challenge in the widespread implementation of DS in the industry. • A significant coding effort may be required to develop a DS. There may also be a need for continued maintenance of the code used for integrating with DS middleware. • The synchronization process between remote services and between visualization and simulation services is far from perfect. • Many of the publications constitute incremental work, which on the one hand demonstrates advancements at the technical level, but on the other hand, ascertains the need for more original, proof of concept and real-world contributions.
RISKS: • Low implementation – not embraced by the industry. • It requires two different kinds of skill-sets—one for developing a computer model and the other for implementing the distributed model. It is a risk to implement technical solutions, without engaging in validation and verification of the computer model itself. Most studies on DSCS focus on message exchange and prevention of causality (akin to a technology-pull solution). • Increase in computing power makes distributed simulation less useful, especially if the goal is faster execution. • In the future, bottom-up analytical approaches such as data mining and process mining could be instrumental in model development. This will be a radical shift from the conventional, top-down approach being employed for the majority of DSCS studies.
RECOMMENDATIONS: • Agenda for future research, as described in section 4.4.3 (refer to Table 9). • Performance enhancements will arise from industrial applications of DSCS frameworks, particularly in highly dynamic multi-enterprise networks (e.g. virtual enterprise). • A prospect in which models and software packages can be combined in a “plug & play” style that utilize the capabilities of a Grid/Cloud.

The new technological era offers DS the potential to meet future challenges by enabling the connectivity and reuse of dispersed models and thereby facilitating the development and execution of large systems simulations. For example, DS can be potentially used for enabling interoperability in Cyber-physical systems (synchronization of physical and software-based systems) and Industry 4.0 applications (synchronization of real-time data from IoT devices with distributed execution of models). Wireless sensor networks and big data can enable simulations to be embedded into operational systems in smart cities [Fujimoto 2016]. Digital Twins as virtual models of real SCs can use symbiotic simulations for monitoring and planning, and DS can enable the distributed execution of digital twins that may represent the different echelons of the supply chains. Moreover, with the increasing adoption of Cloud computing technologies in the industry, Cloud platforms and services could be developed which would enable the SC partners to execute interoperable simulations with databases, visualization, analytics tools, and optimisation models [Anagnostou & Taylor 2017]. Cloud computing can make DS technology widely accessible and relieve users from high purchasing fees for high-performance computing platforms [Fujimoto 2016], thus providing another opportunity for DS to flourish. The question remains, however, of how to make these approaches sustainable. DS research in this area tends to be one-offs and there is little evidence of these innovations being reused in other research or applications. However, it may now be possible to develop sustainable approaches. For example, grid computing approaches have evolved over the past twenty years and have become complex inter-organizational distributed systems or e-Infrastructures (sometimes called cyberinfrastructures) [Bird, Jones and Kee 2009]. These are used to support the needs of different research communities and enables the sharing of common data, simulations, software, sensors, and computing resources used in science. High-performance computing remains a core element of these and use job submission systems such as HTCondor [Bockelman, Livny, Lin and Prelz 2020] to submit jobs across many different types of distributed computing infrastructures such as clusters, networks of PCs, cloud, and high-performance computing facilities [Hey and Trefethen 2005]. A major issue in e-Infrastructures is complex application development and **Workflow Management Systems (WMS)** have emerged to simplify this [Liew et al. 2016]. Given that these systems are typically open source and are supported by large communities, Taylor [2019] argued that these could have a substantial impact on DS and proposed a generic five-stage workflow. It is entirely possible that an e-Infrastructure could be deployed across the organizations of a supply chain. Each organisation would maintain its simulation (or perhaps now its “digital twin”) with links to real-time data. An end user would launch the DSCS workflow. The workflow would acquire and update data feeds and distributions used by the models in the supply chain through links to enterprise systems and sensors managed by edge computing. The models and their updated data would then be composed into federates. These would then be uploaded to cloud or a cluster with a run-time controller (e.g. a RTI). The federation would return results for analysis. Potentially many federations could be executed in parallel for high speed experimentation, depending on available resources. In this new world of digital twins and cyber-physical systems, previous work on symbiotic simulation could be adapted to repeatedly run this workflow to manage the performance of the supply chain in real-time. This, and any other approach to DSCS, needs to take into account the growing number and range of cyberattacks that are happening increasingly in the industry. Lin et al. [2017] and Yan et al. [2013] review a range of cybersecurity threats in IoT and associated areas. Buczak and Guven review data mining and machine learning methods for intrusion detection. Lezzi, Lazoi and Carollo [2018] and Tuptuk and Hailes [2018] review these in the context of cyber-physical systems and Industry 4.0.

6 CONCLUSION

The paper presents the state-of-the-art in distributed supply chain simulation (DSCS) and discusses the agenda for future research. The synthesis of the literature is presented through the framework-based approach (the conceptual ***Profiling, Problems, Models, Outcomes framework*, or *PPMO***), whereby several characteristics of DSCS studies are identified, for example, the context of the application, model development, middleware, integration technologies, implementation, experimental test-beds, and study outcome. The framework was subsequently used to evaluate a total of 131 articles. Full-text reading of the papers captured variables associated with our overarching framework for the study.

The results from this profile of DSCS developments could be useful to a number of stakeholders, such as supply chain analysts, simulation practitioners, distributed systems' programmers, software vendors, and researchers. This study will serve as a reference for those keen on modelling

DSCS and will help them in finding the most appropriate modelling methods, distributed simulation middleware and integration technologies. The efficacy of this work derives not only from the broad observations of this study's findings but also from the issues that are raised and require consideration as research in this field keeps advancing. Through the review of literature, we identified the constraints and trade-offs related to this distributed modelling method, we outlined practical solutions for alleviating the challenges which have led to relatively low adoption of distributed simulation in the context of supply chain management and, finally, we highlighted opportunities for revitalizing DSCS research and practice by fusing time-tested methods and DS algorithms with the emerging new technologies. To conclude, for the researchers, we articulate the need for hybrid modelling and simulation which integrate the existing DS approaches with methods and techniques from other disciplines, for example, Engineering (e.g. Industry 4.0 technologies), Soft Operations Research (e.g. problem conceptualization at the holistic supply chain level) and Software Engineering (e.g., revisiting the lifecycle of a distributed modelling and simulation study with a focus on reusability, agile development and documentation of distributed models). For the SC stakeholders, we emphasize the benefit of applying DSCS for selective sharing of sub-sets of data and models, and which would, on one hand, address the privacy and confidentiality concerns of sensitive data, whilst effectively using non-sensitive data for better management of the overarching supply chain. For example, during a pandemic, such a computational set-up would have helped in better decision making in the context of supply chains for **Personal Protective Equipment (PPE)**, PCR and RPD test kits and vaccines.

REFERENCES

A. Anagnostou and S. J. E. Taylor. 2017. A distributed simulation methodological framework for OR/MS applications. *Simulation Modelling Practice and Theory* 70 (2017), 101–119.

R. Bandinelli, M. Iacono, and A. Orsoni. 2004. Improving the remote scheduling of manufacturing and installation of large custom-made products. In *Proceedings of the 18th European Simulation Multiconference*. 3334–3339.

R. Bandinelli and A. Orsoni. 2005. A DSS simulation model for outsourcing strategies in large-scale manufacturing. In *Proceedings of the 19th European Conference on Modelling and Simulation (ECMS)*. 387–392.

R. Bandinelli, M. Rapaccini, M. Tucci, and F. Visintin. 2006. Using simulation for supply chain analysis: Reviewing and proposing distributed simulation frameworks. *Production Planning & Control* 17 (2006), 167–175.

J. Banks, S. Jain, S. Buckley, P. Lendermann, and M. Manivannan. 2002. Supply chain opportunities: Panel session: Opportunities for simulation in supply chain management. In *Proceedings of the 2002 Winter Simulation Conference*. 1652–1658.

M. Barbuceanu, R. Teigen, and M. S. Fox. 1997. Agent based design and simulation of supply chain systems. In *Proceedings of IEEE 6th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE, 36–41.

M. W. Barnett and C. J. Miller. 2000. Analysis of the virtual enterprise using distributed supply chain modeling and simulation: An application of e-SCOR. In *Proceedings of the 2000 Winter Simulation Conference Proceedings*. IEEE, 352–355.

B. M. Beamon. 1998. Supply chain design and analysis: Models and methods. *International Journal of Production Economics* 55 (1998), 281–294.

I. Bird, B. Jones, and K. F. Kee. 2009. The organization and management of grid infrastructures. *Computer* 42, 1 (2009), 36–46.

B. Bockelman, M. Livny, B. Lin, and F. Prelz. 2020. Principles, technologies, and time: The translational journey of the HTCondor-CE. *Journal of Computational Science*, https://doi.org/10.1016/j.jocs.2020.101213

C. A. Boer. 2005. *Distributed Simulation in Industry. PhD thesis*. Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam, The Netherlands.

C. A. Boer, A. De Brun, and A. Verbraeck. 2009. A survey on distributed simulation in industry. *Journal of Simulation* 3 (2009), 3–16.

S. C. Brailsford, T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio. 2019. Hybrid simulation modelling in operational research: A state-of-the-art review. *European Journal of Operational Research* 278, 3 (2019), 721–737.

A. G. Bruzzone, R. Mosca, A. Orsoni, and R. Revetria. 2002. Simulation-based VV&A methodology for HLA federations: An example from the aerospace industry. In *Proceedings 35th Annual Simulation Symposium*. IEEE, 80–85.

A. G. Bruzzone, R. Mosca, R. Revetria, E. Bocca, and E. Briano. 2005. Agent directed HLA simulation for complex supply chain modeling. *Simulation* 81 (2005), 647–655.

R. E. Bryant. 1977. Simulation of packet communication architecture computer systems (No. MIT/LCS/TR-188). *Massachusetts Institute of Technology, Cambridge.*

A. L. Buczak and E. Guven. 2016. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys and Tutorials* 18, 2 (2016), 1153–1176.

W. Cai, S. J. Turner, and B. P. Gan. 2001. Hierarchical federations: an architecture for information hiding. In *Proceedings of the 15th Workshop on Parallel and Distributed Simulation*. IEEE, 67–74.

K. M. Chandy and J. Misra. 1981. Asynchronous distributed simulation via a sequence of parallel computations. *Communications of the ACM* 24, 4 (1981), 198–206.

R. Chapman and M. Corso. 2005. From continuous improvement to collaborative innovation: The next challenge in supply chain management. *Production Planning & Control* 16 (2005), 339–344.

C. S. Chong, P. Lendermann, B. P. Gan, B. M. Duarte, J. W. Fowler, and T. E. Callarman. 2004. Analysis of a customer demand driven semiconductor supply chain in a distributed simulation test bed. In *Proceedings of the 2004 Winter Simulation Conference*. IEEE, 1902–1909.

C. S. Currie, J. W. Fowler, K. Kotiadis, T. Monks, B. S. Onggo, D. A. Robertson, and A. A. Tako. 2020. How simulation modelling can help reduce the impact of COVID-19. *Journal of Simulation* 14, 2 (2020), 83–97.

J. S. Dahmann, R. M. Fujimoto, and R. M. Weatherly. 1997. The department of defense high level architecture. In *Proceedings of the 1997 Winter Simulation Conference*. IEEE, 142–149.

C. Dan, S. J. Turner, and W. Cai. 2008. Towards fault-tolerant HLA-based distributed simulations. *Simulation* 84 (10–11), 493–509.

M. A. Dalal, H. Bell, M. Denzien, and M. P. Keller. 2003. Supply chain management simulation: Initializing a distribution supply chain simulation with live data. In *Proceedings of the 2003 Winter Simulation Conference*. IEEE, 1621–1626.

P. Davidsson. 2000. Multi-agent-based simulation: beyond social simulation. In *Multi-Agent-Based Simulation*. Springer, 97–107.

Y. K. Dwivedi, B. Lal, N. Mustafee, and M. D. Williams. 2009. Profiling a decade of information systems frontiers' research. *Information Systems Frontiers* 11, 1 (2009), 87–102.

O. Eddama and J. Coast. 2008. A systematic review of the use of economic evaluation in local decision-making. *Health Policy* 86 (2008), 129–141.

B. Elahi, A. W. Malik, A. U. Rahman, and M. A. Khan. 2020. Toward scalable cloud data center simulation using high-level architecture. *Software: Practice and Experience* 50, 6 (2020), 827–843.

Z. Fanchao, S. J. Turner, and H. Aydt. 2009. Symbiotic simulation control in supply chain of lubricant additive industry. In *the 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*. IEEE, 165–172.

M. Fayez, L. Rabelo, and M. Mollaghasemi. 2005. Ontologies for supply chain simulation modeling. In *Proceedings of the 2005 Winter Simulation Conference*. IEEE, 2364–2370.

R. M. Fujimoto. 1990. Parallel discrete event simulation. *Communications of the ACM* 33, 10 (1990), 30–53.

R. M. Fujimoto. 2000. *Parallel and Distributed Simulation Systems*. Wiley, New York, NY.

R. M. Fujimoto. 2015. Parallel and distributed simulation. In *Proceedings of the 2015 Winter Simulation Conference*. IEEE, 45–59.

R. M. Fujimoto. 2016. Research challenges in parallel and distributed simulation. *ACM Transactions on Modeling and Computer Simulation* 26, 4 (2016), 1–29.

B. P. Gan and S. J. Turner. 2000. An asynchronous protocol for virtual factory simulation on shared memory multiprocessor systems. *Journal of the Operational Research Society* 51, 4 (2000), 413–422.

B. P. Gan, J. Wei, X. Wang, and W. Cai. 2003. Synchronization and management of shared state in HLA-based distributed simulation. In *Proceedings of the 2003 Winter Simulation Conference*. IEEE, 847–854.

B. P. Gan, L. Liu, S. Jain, S. J. Turner, W. Cai, and W. J. Hsu. 2000a. Distributed supply chain simulation across enterprise boundaries. In *Proceedings of the 2000 Winter Simulation Conference*. IEEE, 1245–1251.

B. P. Gan, Y. H. Low, S. Jain, S. J. Turner, W. Cai, W. J. Hsu, and S. Y. Huang. 2000b. Load balancing for conservative simulation on shared memory multiprocessor systems. In *Proceedings of the 14th Workshop on Parallel and Distributed Simulation*. IEEE, 139–146.

P. Gao, C. Jin, and G. Wang. 2008. HLA-based distributed simulation model for multimodal operation system on container terminals. In *Proceedings of the 2008 Asia Simulation Conference*. IEEE, 338–343.

N. Gilbert. 2019. Agent-based models (Vol. 153). *Sage Publications*.

M. M. Gunal. (Ed.). 2019. *Simulation for Industry 4.0*. Cham, Switzerland: Springer.

T. Hey and A. E. Trefethen. 2005. Cyberinfrastructure for e-Science. *Science* 308, 5723, 817–821.

J. Hongyu, L. Xia, and C. Yan. 2010. The research of a distributed simulation method of information sharing in supply chain. In *Proceedings of the 2002 International Conference on Optoelectronics and Image Processing*. IEEE, 596–599.

IEEE. 2010. *IEEE 1516-2011 IEEE Standard for Modeling and Simulation (M&S) High-level Architecture (HLA)*. New York, NY: Institute of Electrical and Electronics Engineer.

P. H. M. Jacobs. 2005. *The DSOL Simulation Suite. Ph. D. thesis*, Delft University of Technology, Delft, The Netherlands.

P. H. Jacobs, N. A. Lang, and A. Verbraeck. 2002. D-SOL: A distributed Java based discrete event simulation architecture. In *Proceedings of the 2002 Winter Simulation Conference*. IEEE, 793–800.

S. Jain, F. Riddick, A. Craens, and D. Kibira. 2007. Distributed simulation for interoperability testing along the supply chain. In *Proceedings of the 2007 Winter Simulation Conference*. IEEE, 1044–1052.

D. Jeong, M. Seo, and Y. Seo. 2009. Development of web-based simulator for supply chain management. In *Proceedings of the 2009 Winter Simulation Conference*. IEEE, 2303–2309.

F. Jia, X. Wang, N. Mustafee, and L. Hao. 2016. Investigating the feasibility of supply chain-centric business models in 3D chocolate printing: A simulation study. *Technological Forecasting and Social Change* 102 (2016), 202–213.

W. Jian, H. Yang, and W. ZiYang. 2017. Research on modeling and simulation of distributed supply chain based on HAS. In *Proceedings of the 2017 International Conference on Intelligent Robotics and Applications*. Springer, Cham, 401–412.

O. Jellouli and E. Chatelet. 2001. Monte carlo simulation and genetic algorithm for optimizing supply chain management in a stochastic environment. In *Proceedings of the 2001 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 1835–1839.

K. Katsaliaki and S. C. Brailsford. 2007. Using simulation to improve the blood supply chain. *Journal of the Operational Research Society* 58 (2007), 219–227.

K. Katsaliaki, N. Mustafee, S. J. E. Taylor, and S. Brailsford. 2009. Comparing conventional and distributed approaches to simulation in complex supply-chain health systems. *Journal of the Operational Research Society* 60 (2009), 43–51.

K. Katsaliaki and N. Mustafee. 2010. Improving decision making in healthcare services through the use of existing simulation modelling tools and new technologies. *Transforming Government: People, Process and Policy* 4 (2010), 158–171.

F. Kuhl, R. Weatherly, and J. Dahmann. 1999. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Prentice Hall PTR.

K. Klobut, J. Shemeikka, P. Tuomaala, K. Sipilä, and J. Heikkinen. 2007. ENTRY - Management of distributed energy systems. In *Intelligent Products and Systems. Technology theme - Final Report*. Ventä, O. (Ed.), VTT Publications.

K. Kotiadis, A. A. Tako, and C. Vasilakis. 2014. A participative and facilitative conceptual modelling framework for discrete event simulation studies in healthcare. *Journal of the Operational Research Society* 65, 2 (2014), 197–213.

R. Iannone, S. Miranda, and S. Riemma. 2007. Supply chain distributed simulation: An efficient architecture for multi-model synchronization. *Simulation Modelling Practice and Theory* 15, 3 (2007), 221–236.

D. R. Jefferson. 1985. Virtual time. *ACM Transactions on Programming Languages and Systems* 7, 3 (1985), 404–425.

Z. Ji, B. P. Gan, S. J. Turner, and W. Cai. 2001. Parallel federates-an architecture for hybrid distributed simulation. In *Distributed Simulation and Real Time Applications*. IEEE, 97–104.

D. M. Lambert and M. C. Cooper. 2000. Issues in supply chain management. *Industrial Marketing Management* 29 (2000), 65–83.

H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann. 2014. Industry 4.0. *Business & Information Systems Engineering* 6, 4 (2014), 239–242.

A. M. Law and W. D. Kelton. 2000. *Simulation Modeling and Analysis* (Vol. 3). New York: McGraw-Hill.

J.-H. Lee and C.-O. Kim. 2008. Multi-agent systems applications in manufacturing systems and supply chain management: A review paper. *International Journal of Production Research* 46 (2008), 233–265.

S. Lee, Y.-J. Son, and R. A. Wysk. 2007. Simulation-based planning and control: From shop floor to top floor. *Journal of Manufacturing Systems* 26, 2 (2007), 85–98.

P. Lendermann. 2006. About the need for distributed simulation technology for the resolution of real-world manufacturing and logistics problems. In *Proceedings of the 2006 Winter Simulation Conference*. IEEE, 1119–1128.

P. Lendermann, B. P. Gan, and L. F. McGinnis. 2001. Distributed simulation with incorporated APS procedures for high-fidelity supply chain optimization. In *Proceeding of the 2001 Winter Simulation Conference*. IEEE, 1138–1145.

P. Lendermann, N. Julka, B. P. Gan, D. Chen, L. F. McGinnis, and J. P. McGinnis. 2003. Distributed supply chain simulation as a decision support tool for the semiconductor industry. *Simulation* 79, 3 (2003), 126–138.

P. Lendermann, M. U. Heinicke, L. F. McGinnis, C. Mclean, S. Strassburger, and S. J. E. Taylor. 2007. Panel: Distributed simulation in industry-a real-world necessity or ivory tower fancy? In *Proceedings of the 2007 Winter Simulation Conference*. IEEE, 1053–1062.

K. Lepenioti, A. Bousdekis, D. Apostolou, and G. Mentzas. 2020. Prescriptive analytics: Literature review and research challenges. *International Journal of Information Management* 50 (2020), 57–70.

M. Lezzi, M. Lazoi, Corallo. 2018. Cybersecurity for industry 4.0 in the current literature: A reference framework. *Computers in Industry* 103 (2018), 97–110.

T. Li, Y. Yao, W. Tang, F. Yao, and F. Zhu. 2019. An exponential search enhanced dynamic sort-based interest matching algorithm for interest management in distributed simulation. *Simulation Modelling Practice and Theory* 95 (2019), 78–95.

X. Li, W. Mao, D. Zeng, and F. Y. Wang. 2008. Agent-based social simulation and modeling in social computing. In *International Conference on Intelligence and Security Informatics*. Springer, Berlin, 401–412.

C. S. Liew, M. P. Atkinson, M. Galea, T. F. Ang, P. Martin, and J. L. Van Hemert. 2016. Scientific workflows: Moving across paradigms. *ACM Computing Surveys* 49, 4 (2016), 1–39. https://doi.org/10.1145/3012429

J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. 2017. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal* 4, 5 (2017), 1125–1142.

R. J. Linn, C. S. Chen, and J. A. Lozan. 2002. Development of distributed simulation model for the transporter entity in a supply chain process. In *Proceedings of the 2002 Winter Simulation Conference*. IEEE, 1319–1326.

Q. Long, J. Lin, and Z. Sun. 2011. Modeling and distributed simulation of supply chain with a multi-agent platform. *The International Journal of Advanced Manufacturing Technology* 55, 9–12, 1241–1252.

M. Y. H. Low and S. J. Turner. 2006. Using web services and business process execution language for HLA-based distributed supply-chain simulation. In *Proceedings of the 2006 UK Operational Research Society Simulation Workshop*. UK OR Society, 23–31.

M. Marzolla and G. D'angelo. 2020. Parallel data distribution management on shared-memory multiprocessors. *ACM Transactions on Modeling and Computer Simulation* 30, 1 (2020), 1–25.

L. F. McGinnis. 2004. Distributing a large-scale, complex FAB simulation using HLA and Java: Issues and lessons. In *Proceedings of the 2004 Winter Simulation Conference*. IEEE, 1891–1895.

C. McLean and F. Riddick. 2001. Integrating distributed manufacturing simulations. In *Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 1294–1298.

L. Mekaouche, F. Ounnar, P. Pujo, and N. Giambiasi. 2009. Another way to manage supply chains: Holonic and multicriteria approach. *International Journal of Logistics Systems and Management*. Inderscience.

K. Mertins and M. Rabe. 2005. Inter-enterprise planning of manufacturing systems applying simulation with IPR protection. In Arai E., Kimura F., Goossenaerts J., Shirase K. (Eds.) *Knowledge and Skill Chains in Engineering and Manufacturing*. DIISM 2002. IFIP International Federation for Information Processing, vol 168. Springer, Boston, MA.

K. Mertins, M. Rabe, and F.-W. Jakel. 2005. Distributed modelling and simulation of supply chains. *International Journal of Computer Integrated Manufacturing* 18 (2005), 342–349.

D. C. Miller and J. A. Thorpe. 1995. SIMNET: The advent of simulator networking. *Proceedings of the IEEE* 83, 8 (1995), 1114–1123.

J. Misra. 1986. Distributed discrete-event simulation. *ACM Computing Surveys* 18, 1 (1986), 39–65.

C. Z. Mooney. 1997. *Monte Carlo Simulation*. Sage Publications.

T. Moyaux, B. Chaib-Draa, and S. d'Amours. 2006. Supply chain management and multiagent systems: An overview. In *Multiagent based Supply Chain Management Springer*. 1–27.

N. Mustafee. 2004. *Performance Evaluation of Interoperability Methods for Distributed Simulation. MSc. thesis*. Brunel University, London, UK.

N. Mustafee, S. J. E. Taylor, K. Katsaliaki, and S. Brailsford. 2009. Facilitating the analysis of a UK national blood service supply chain using distributed simulation. *Simulation* 85, 2 (2009), 113–128.

N. Mustafee, K. Katsaliaki, and S. J. Taylor. 2010. Profiling literature in healthcare simulation. *Simulation* 86, 8–9 (2010), 543–558.

N. Mustafee, K. Katsaliaki, P. Fishwick, and M. D. Williams. 2012. SCS: 60 years and counting! A time to reflect on the society's scholarly contribution to M&S from the turn of the millennium. *Simulation* 88, 9 (2012), 1047–1071.

N. Mustafee, S. J. E. Taylor, K. Katsaliaki, Y. Dwivedi, and M. Williams. 2012. Motivations and barriers in using distributed supply chain simulation. *International Transactions in Operational Research* 19 (2012), 733–751.

N. Mustafee, K. Katsaliaki, and S. J. E. Taylor. 2014. A review of literature in distributed supply chain simulation. In *Proceedings of the 2014 Winter Simulation Conference*. IEEE, 2872–2883.

N. Mustafee, M. Sahnoun, A. Smart, and P. Godsiff. 2015. An application of distributed simulation for hybrid modeling of offshore wind farms. In *Proceedings of the 2015 ACM SIGSIM/PADS Conference on Principles of Advanced Discrete Simulation*. ACM, 171–172.

D. Nicol and P. Heidelberger. 1996. Parallel execution for serial simulators. *ACM Transactions on Modeling and Computer Simulation* 6 (1996), 210–242.

P. Nilsen. 2015. Making sense of implementation theories, models and frameworks. *Implementation Science* 10 (2015), 53–79.

J. M. Nurmilaakso. 2004. Supply chain scheduling using distributed parallel simulation. *Journal of Manufacturing Technology Management* 15, 8 (2004), 756–770.

I. B. Owusu and S. Hauan. 2006. Information sharing in a distributed enterprise: Impact on supply chain performance and decision-making. *Computer Aided Chemical Engineering* 21, (B) (2006), 2195.

K. Pan, S. J. Turner, W. Cai, and Z. Li. 2007. A service oriented HLA RTI on the grid. In *Proceedings of the IEEE International Conference on Web Service*. IEEE, 984–992.

S. D. Pathak, D. M. Dilts, and G. Biswas. 2004. Simulating growth dynamics in complex adaptive supply networks. In *Proceedings of the 2004 Winter Simulation Conference*. IEEE, 774–782.

M. H. Penedo. 1989. Acquiring experiences with executable process models. In *Proceedings of the 5th International Software Process Workshop*. IEEE Computer Society, 112–113.

E. Pereira, M. Ali, L. Wu, and S. Abourizk. 2020. Distributed simulation-based analytics approach for enhancing safety management systems in industrial construction. *Journal of Construction Engineering and Management* 146, 1 (2020), 04019091.

A. L. Porter, A. Kongthon, and J. C. J. Lu. 2002. Research profiling: Improving the literature review. *Scientometrics* 53, 3 (2002), 351–370.

E. Proctor, H. Silmere, R. Raghavan, P. Hovmand, G. Aarons, A. Bunger, and M. Hensley. 2011. Outcomes for implementation research: Conceptual distinctions, measurement challenges, and research agenda. *Administration and Policy in Mental Health and Mental Health Services Research* 38, 2 (2011), 65–76.

M. Rabe and F. W. Jaekel. 2003. The MISSION Project. In Kosanke K., Jochem R., Nell J. G., Bas A. O. (Eds.) *Enterprise Inter- and Intra-Organizational Integration*. In *Proceedings of the 2002 International Conference on Enterprise Integration and Modeling Technology (ICEIMT). IFIP — The International Federation for Information Processing*, vol 108. Springer, Boston, MA.

M. Rabe, F. W. Jaekel, and H. Weinaug. 2006. Reference models for supply chain design and configuration. In *Proceedings of the 2006 Winter Simulation Conference*. IEEE, 1143–1150.

A. Rathore, B. Balaraman, X. Zhao, J. Venkateswaran, Y. J. Son, and R. A. Wysk. 2005. Development and benchmarking of an epoch time synchronization method for distributed simulation. *Journal of Manufacturing Systems* 24, 2 (2005), 69–78.

R. Revetria, P. E. J. N. Blomjous, and S. P. A. Van Houten. 2003. An HLA federation for evaluating multi-drop strategies in logistics. In *Proceedings of the 15th European Simulation Symposium and Exhibition Conference*. European Council for Modelling and Simulation, 450–455.

S. Robinson. 2005. Discrete-event simulation: from the pioneers to the present, what next? *Journal of the Operational Research Society* 56 (2005), 619–629.

S. Robinson. 2008. Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society* 59, 3 (2008), 278–290.

M. D. Rossetti and Y. Chen. 2012. A cloud computing architecture for supply chain network simulation. In *Proceedings of the 2012 Winter Simulation Conference*. IEEE, 1–12.

S. M. Saad, T. Perera, and R. Wickramarachchi. 2003. Simulation of distributed manufacturing enterprises: A new approach. In *Proceedings of the 2003 Winter Simulation Conference*. IEEE, 1167–1173.

SISO. 2010. Standard for commercial-off-the-shelf simulation package interoperability reference models (SISO-STD-006-2010). *Simulation Interoperability Standards Organization*. https://www.sisostds.org/ProductsPublications/Standards/SISOStandards.aspx (last accessed May 2020).

I. Sommerville. 1996. Software process models. *ACM computing surveys (CSUR)* 28, 1 (1996), 269–271.

Y.-J. Son. 2005. Distributed simulation for multi-university collaborative manufacturing. In *Proceedings of the 2005 International Conference on Manufacturing Education*, San Luis Obispo, CA.

J. D. Sterman. 2001. System dynamics modeling: Tools for learning in a complex world. *California Management Review* 43 (2001), 8–25.

G. C. Stevens. 1989. Integrating the supply chain. *International Journal of Physical Distribution & Materials Management* 19 (1989), 3–8.

R. Sudra, S. J. Taylor, and T. Janahan. 2000. Distributed supply chain simulation in GRIDS. In *Proceedings of the 2000 Winter Simulation Conference Proceedings*. IEEE, 356–361.

S. Tammineni and J. Venkateswaran. 2007. Advanced look-ahead based approach (ALBA) for distributed simulation of supply chains. In *Proceedings of the 2007 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, 1838–1842.

N. Tampouratzis, I. Papaefstathiou, A. Nikitakis, A. Brokalakis, S. Andrianakis, A. Dollas, and E. Plebani. 2020. A novel, highly integrated simulator for parallel and distributed systems. *ACM Transactions on Architecture and Code Optimization* 17, 1 (2020), 1–28.

G. Tan, N. Zhao, and S. J. E. Taylor. 2003. Automobile manufacturing supply chain simulation in the GRIDS environment. In *Proceedings of the 2003 Winter Simulation Conference*. IEEE, 1149–1157.

S. J. E. Taylor. 2019. Distributed simulation: State-of-the-art and potential for operational research. *European Journal of Operational Research* 273, 1 (2019), 1–19.

S. J. E. Taylor, A. Bruzzone, R. Fujimoto, B. P. Gan, S. Strassburger, and R. Paul. 2002a. Distributed simulation and industry: Potentials and pitfalls. In *Proceedings of the 2002 Winter Simulation Conference*. IEEE, 688–694.

S. J. E. Taylor, R. Sudra, T. Janahan, G. Tan, and J. Ladbrook. 2002b. GRIDS-SCF: An infrastructure for distributed supply chain simulation. *Simulation* 78, 5 (2002), 312–320.

S. J. E. Taylor, S. Straßburger, B. P. Gan, and A. Verbraeck. 2003. HLA-CSPIF panel on commercial off-the-shelf distributed simulation. In *Proceedings of the 2003 Winter Simulation Conference*. IEEE, 881–887.

S. J. E. Taylor, T. Eldabi, G. Riley, R. J. Paul, and M. Pidd. 2009a. Simulation modelling is 50! Do we need a reality check? *Journal of the Operational Research Society* 60 (2009a), S69–S82.

S. J. E. Taylor, N. Mustafee, S. J. Turner, K. Pan, and S. Straßburger. 2009b. Commercial-off-the-shelf simulation package interoperability: Issues and futures. In *Proceedings of the 2009 Winter Simulation Conference*. IEEE, 203–215.

S. J. E. Taylor, R. Fujimoto, E. H. Page, P. A. Fishwick, A. M. Uhrmacher, and G. Wainer. 2012a. Panel on grand challenges for modeling and simulation. In *Proceedings of the 2012 Winter Simulation Conference*. IEEE, 232.

S. J. E. Taylor, S. J. Turner, S. Straßburger, and N. Mustafee. 2012b. Bridging the gap: A standards-based approach to OR/MS distributed simulation. *ACM Transactions on Modeling and Computer Simulation* 22, Article 18.

S. Terzi and S. Cavalieri. 2004. Simulation in the supply chain context: A survey. *Computers in Industry* 53, (2004), 3–16.

K. D. Tocher. 1963. *The art of simulation*. English Universities Press, London.

A. Tolk. 2012. *Engineering Principles of Combat Modeling and Distributed Simulation*. John Wiley & Sons, New York, NY.

A. Tolk, C. Glazner, and J. Ungerleider. 2021. Computational decision support for the COVID-19 healthcare coalition. *Computing in Science & Engineering* 23, 1 (2021), 17–24.

N. Tuptuk and S. Hailes. 2018. Security of smart manufacturing systems. *Journal of Manufacturing Systems* 47, (2018), 93–106.

S. J. Turner, W. Cai, and B. P. Gan. 2000. Adapting a supply-chain simulation for HLA. In *Proceedings Fourth IEEE International Workshop on Distributed Simulation and Real-Time Applications (DS-RT 2000)*. IEEE, 71–78.

S. P. van Houten and P. H. Jacobs. 2004. An architecture for distributed simulation games. In *Proceedings of the 2004 Winter Simulation Conference*. IEEE, 2081–2086.

R. van Krevelen, M. Warnier, F. Brazier, A. Verbraeck, and T. Corsi. 2011. Transparency, consistency and modularity of strategic reasoning: An agent architecture for interactive business simulations. In *Proceedings of the 2011 Winter Simulation Conference*. IEEE, 1612–1623.

J. Venkateswaran and Y.-J. Son. 2004. Design and development of a prototype distributed simulation for evaluation of supply chains. *International Journal of Industrial Engineering: Theory Applications and Practice* 11, (2004), 151–160.

J. Venkateswaran and Y.-J. Son. 2005. Hybrid system dynamic—discrete event simulation-based architecture for hierarchical production planning. *International Journal of Production Research* 43, (2005), 4397–4429.

J. Venkateswaran and Y.-J. Son. 2009. Robust supply-chain planning using multi-resolution hybrid models: Experimental study. *International Journal of Modelling and Simulation* 29, (2009), 417–427.

J. Venkateswaran, X. Zhao, and Y.-J. Son. 2005. Integrated performance and stability analysis for VMI supply chains using multi-resolution hybrid models. In *Proceedings of the 2005 IIE Annual Conference*. Institute of Industrial Engineers.

C. Wartha, M. Peev, A. Borshchev, and A. Filippov. 2002. Decision support tool-supply chain. In *Proceedings of the 2002 Winter Simulation Conference*. IEEE, 1297–1301.

A. L. Wilson and R. M. Weatherly. 1994. The aggregate level simulation protocol: An evolving system. In *Proceedings of the 1994 Winter Simulation Conference. Society for Computer Simulation International (SCS)*, 781–787.

Y. Wu, L. Fu, F. Ma, and X. Hao. 2020. Cyber-physical co-simulation of shipboard integrated power system based on optimized event-driven synchronization. *Electronics* 9, 3 (2020), 540.

X. Xu and J. Lin. 2009. A novel time advancing mechanism for agent-oriented supply chain simulation. *Journal of Computers* 4, 12 (2009), 1301–1308.

Y. Yan, Y. Qian, H. Sharif, and D. Tipper. 2013. A survey on smart grid communication infrastructures: Motivations, requirements and challenges. *IEEE Communications Surveys and Tutorials* 15, 1 (2013), 5–20.

M. J. Yoo and R. Glardon. 2009. Combining JADE and repast for the complex simulation of enterprise value-adding networks. In *Proceedings of the 2008 International Workshop on Agent-Oriented Software Engineering*. Springer, Berlin, 243–256.

G. Zacharewicz, J. C. Deschamps, and J. Francois. 2011. Distributed simulation platform to design advanced RFID based freight transportation systems. *Computers in Industry* 62, 6 (2011), 597–612.

B. P. Zeigler and D. Kim. 1999. Distributed supply chain simulation in a DEVS/CORBA execution environment. In *Proceedings of the 1999 Winter Simulation Conference*. IEEE, 1333–1340.

S. Zhou, D. Chen, W. Cai, L. Luo, M. Y. H. Low, F. Tian, V. S.-H. Tay, D. W. S. Ong, and B. D. Hamilton. 2010. Crowd modeling and simulation technologies. *ACM Transactions on Modeling and Computer Simulation* 20, 4 (2010), 1–35.

Received July 2020; revised March 2021; accepted May 2021

Author Queries

Q1: AU: Please supply the CCS Concepts 2012 codes per the ACM style indicated on the ACM website Please include the CCS Concepts XML coding as well.

Q2: AU: Do you mean “iterations” or “interactions”?