Simulation Modelling Practice and Theory 70 (2017) 101–119

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

A distributed simulation methodological framework for OR/MS applications

Anastasia Anagnostou*, Simon J.E. Taylor

Department of Computer Science, Brunel University London, UK

ARTICLE INFO

Article history:
Received 31 May 2016
Revised 14 October 2016
Accepted 24 October 2016

Keywords:
Distributed Simulation
HLA
DSEEP
Hybrid Simulation
Discrete Event Simulation
Agent-Based Simulation
Operations Research and Management Science
Simulation methodology

ABSTRACT

Distributed Simulation (DS) allows existing models to be composed together to form simulations of large-scale systems, or large models to be divided into models that execute on separate computers. Among its claimed benefits are model reuse, speedup, data privacy and data consistency. DS is arguably widely used in the defence sector. However, it is rarely used in Operations Research and Management Science (OR/MS) applications in areas such as manufacturing and healthcare, despite its potential advantages. The main barriers to use DS in OR/MS are the technical complexity in implementation and a gap between the world views of DS and OR/MS communities. In this paper, we propose a new method that attempts to link together the methodological practices of OR/MS and DS. Using a representative case study, we show that our methodological framework simplifies significantly DS implementation.

1. Introduction

For many decades, researchers and practitioners in the field of Operations Research and Management Science (OR/MS) have used Modelling and Simulation (M&S) for systems analysis and decision making. The power of M&S is the ability to represent a real system and to study the impact of changes without actually altering the physical system. It provides insight into operational processes and can be used for dynamic forecasting, what-if analysis and optimisation at strategic, tactical and operational levels planning. OR/MS M&S is used in a wide range of areas including commerce, healthcare, manufacturing and logistics and covers a range of techniques from System Dynamics (SD) to Discrete Event Simulation (DES) and, more recently, Agent-Based Simulation (ABS). These techniques are supported by simulation software packages offering complete simulation environments, such as Visual Interactive Modelling Systems (VIMS) that support all (or most) aspects of a simulation study. VIMS typically run on a desktop computer and provide a user-friendly modelling interface that can be used with a minimal knowledge of programming. Models are easily developed by the interface and support a wide range of system elements and attributes. Sophisticated visualisation and results analysis tools support simulation experimentation and reporting.

However, OR/MS M&S practices have some limitations, especially when large-scale systems are being modelled. A single simulation run of a model can take a long time and can therefore limit the amount of experimentation performed in a project. Arguably, larger models take longer to simulate. Indeed, in some cases, when a model is considerably large

* Corresponding author.
E-mail addresses: anastasia.anagostou@brunel.ac.uk (A. Anagnostou), simon.taylor@brunel.ac.uk (S.J.E. Taylor).

http://dx.doi.org/10.1016/j.simpat.2016.10.007
1569-190X/© 2016 Published by Elsevier B.V.

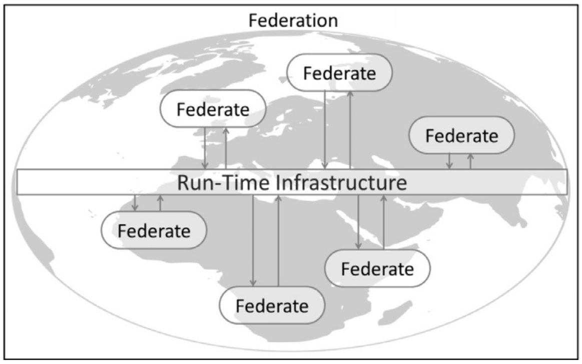

Fig. 1. HLA distributed simulation system.

and involves many objects and events, significant compromises must be made in experimentation when using a single desktop computer [24]. Further, large-scale systems usually involve subsystems either within the same organisation or in autonomous organisations with interdependencies. It may be convenient to create a hybrid model that uses different M&S techniques to represent different systems [8]. The typical approach to building OR/MS models is to use a single simulation package running on a single computer. Certain compromises are necessary in order to deal with the size and heterogeneity of large systems. In order to be able to simulate a large model in reasonable time, OR/MS modellers typically increase the level of abstraction at the cost of detail included in the model. This can lead to a loss of accuracy of results. Furthermore, as most simulation packages only support a single M&S technique, a model is typically developed using a single simulation technique (currently only one commercial simulation tool supports multi-technique modelling, Anylogic, www.anylogic.com). The need to use hybrid approaches is becoming more apparent as, for example, enterprise operations are becoming more interoperable [52]. Modern enterprises operate in networks, moving more and more into increasingly interdependent processes. Consequently, OR/MS modellers have to deal with more complicated, heterogeneous and larger processes. In summary, how can OR/MS modellers address larger and larger systems without compromising accuracy, using convenient hybrid approaches and within a reasonable experimentation time frame?

The practice of M&S in defence and space exploration has evolved on a different path to OR/MS. Applications of M&S in these areas related to training has focussed on the development of real-time simulations (for warfare training or space vehicle operations) [40, 35]. Motivated by the high cost of these simulations and the need to reuse models essentially as components of larger systems, much work, since the early 1990 s, has focussed on Distributed Simulation (DS) and simulation interoperability. One of the major outcomes of this work is the standardisation of the way models interoperate and has led to the well-known High-Level Architecture (HLA) standard [21]. The impact of this standard and its widespread adoption in the above areas is that it is normal for models to be built as interoperating components of large scale models. As shown in Fig. 1, logically models interact directly as if they were part of a single model. Physically models run on different computers (potentially located across the world) and interact over a communications network via software (middleware), based on the HLA standard, called the Run-Time Infrastructure (RTI). The HLA has its own terminology that can be quite opaque to “outsiders” – each model is a federate and a collection of models is a federation. Federates can be recomposed to form different federations. Each federate executes independently and can select what information will be shared with the other federates in the federation.

It appears that the M&S approaches made possible by the HLA could solve large model issues in OR/MS. Models could be composed of submodels that run on the resources of their own computers with no need to sacrifice detail. Similarly, models could be created in different simulation software supporting different simulation techniques, as appropriate. These models could be linked together as subsystems of larger models. This concept of using the HLA to support large-scale OR/MS simulations is not new. Clearly, distributed simulation could be employed by OR/MS as a viable simulation approach for large-scale interoperable systems. Characteristics such as sharing the computational load over a network and building independent, interoperable simulations can facilitate the issues mentioned above. Subsystems of the whole system can be modelled as federates using the most appropriate simulation technique. These federates can communicate and share data at an interface level while keeping their individuality and privacy. The interoperable federates can be reused in other large-scale systems. Finally, each federate can be executed in a network node locally or remotely. Therefore, DS could provide a solution to inadequate computing power by distributing the load over a network, to heterogeneity by forming hybrid federations and therefore capturing realistically the whole system, to inter-organisation information sharing and privacy by selecting the level of transparency, and to model reusability by developing interoperable and composable models.

However, despite over a decade of research into approaches to create these OR/MS DS, only a very small number of industrial examples are documented (see Section 2). We argue that the main reason for this is not a lack of demand but the complexity and opaqueness of the HLA to OR/MS practitioners and researchers. As identified in Hannay et al. [15], there is a lack in software engineering training for academic researchers and application practitioners. Scientists do not have a clear understanding of software engineering processes. However, the need for formal processes and further software engineering training is important as the software tools become more complex. In this paper, we propose a new method that attempts to link together the methodological practices of OR/MS and the HLA. Our goal is to extend OR/MS practice with clearly defined and, where appropriate, translated elements of the HLA to enable the creation of large-scale models. We, therefore, present a DS methodological framework for OR/MS applications that attempts to bridge the chasm between the two world views. By identifying the corresponding practices of two of the most well-known simulation life-cycle frameworks for OR/MS and DS worlds, namely the OR/MS simulation method by Banks et al. [5] and the IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP) standard [18], we propose our DS methodological framework using notation familiar and widely acceptable by OR/MS modellers and translating notation and terms familiar only to the software and systems engineering world.

The rest of the paper is organised as follows. In Section 2, there is a discussion on the background literature on DS works in OR/MS domain. Then, Section 3 discusses the development process of the proposed OR/MS DS methodological framework and explains the involved activities in respect to OR/MS and DS world views. Section 4 discusses the detailed steps of the proposed framework. Section 5 presents a case study implementation applying the OR/MS DS methodological framework on an illustrative example of hybrid ABS-DES. Reflections on the way the proposed approach eased the development of the OR/MS DS case study as well as its limitations and suggestions for improvements are discussed in Section 6. Section 7 concludes the paper.

2. Background literature

DS is widely used in the defence sector and some areas of the space industry (led by NASA). As noted below there are some examples of successful DS applications in other areas. However, concerns raised by Strassburger et al. [46] and Boer et al.'s survey of 2004 [7], where experts in the field of industrial M&S applications commented on the DS difficulties seen by industry, note issues that might limit the widespread adoption of DS. These include the cost of RTI software, the complexity of DS, the lack of DS experts in industry, and the lack of guidance in including the functionalities of DS standards to industrial applications. More recently Fujimoto [12] notes the challenge of making DS more accessible and the complexity of creating these applications. This limited adoption is further supported by a survey conducted by Fakhimi and Probert [11] that reviewed the academic literature of OR/MS techniques applied in the UK healthcare sector. They found only three publications that applied DS. These appear after the year 2006 and involve a single case study. It must be noted that there may be other non-defence DS success stories developed in industry that have not been reported in primary or secondary literature.

Potentially, industry has a lot to gain from DS especially for large-scale simulation projects. With the advent of cloud computing, M&S could further benefit from "unlimited" on-demand distributed computing power. How can we encourage OR/MS modellers to overcome the above barriers to adoption? One first step is to start talking in their "language". Instead of expecting OR/MS modellers to lean toward the software engineering practices of DS, we can translate these practices into ones convenient and familiar to OR/MS modellers.

There are few studies in the academic literature that employed DS to simulate applications in the OR/MS area. For example, in the late 1990 s, Klein et al. [25] developed a HLA-based DS for traffic simulation. They reused existing models that were extended to include HLA interfaces. The focus was on interoperability and reusability aspects of HLA and therefore they did not comment of RTI performance or development effort. They achieved platform, language and time management interoperability by using different simulation and animations tools, linking scaled real-time and event-oriented simulations. DS traffic simulation functionality can greatly increase by interoperable Geographic Information Systems (GIS) modules, a DS application that is achieved by Bernard et al. [6]. In the context of emergency management, they developed an HLA-based GIS component to represent the dynamic nature of the system. Evidently, by such interoperable simulation components that use common standards, reusability is supported. For example, the same HLA GIS can be used in various DS federations. Hahn et al. [14] developed an HLA-based DS suite for maritime systems. Their system, HAGGIS, includes heterogeneous models and simulations such as semantic models, maritime traffic simulations and sensor simulations that can interoperate via the RTI implementation.

Klein et al. [25] and Bernard et al. [6] created DS of DES models and their peripheral services. In another OR/MS field, that of managing large construction projects, Taghaddos et al. [48] developed an application for HLA ABS and Multi-Agent System (MAS) simulations. They also included scheduling and optimisation. These are independent federates that can be recomposed to form different federations. Again, they stress the need for breaking down large-scale systems into subsystems and modelling them independently for more efficient experimentations and model recycling.

One OR/MS area that is strongly considered for DS is Supply Chain Management (SCM). A supply chain consists of a large network of independent organisations, that must cooperate closely but be coupled loosely, each of which could be a separate model. The majority of OR/MS articles on DS report on some form of supply chain. Bruzzone et al. [9] developed an HLA ABS federation for a country-wide supply chain in Italy where federates are able to negotiate, react and reschedule events dynamically. The transportation network also constitutes a federate. Furthermore, they raised the issue of lack of communication security in HLA and they applied a security protocol to authenticate users. DES DS was applied by Jain et al. [22] and Katsaliaki et al. [24]. They modelled complex supply chain operations in manufacturing and healthcare, respectively. Jain et al. [22] proposed a DS for testing interoperability standards and compliance of applications in supply chain information systems. With an example from the automotive industry, they created a virtual manufacturing environment using the Distributed Manufacturing Simulation Adapter (DMS), developed by the National Institute of Standards and Technology (NIST). The DMS Adapter is a simplified implementation of the HLA standard. The DS includes federates from the heterogeneous supply chain up to processes on the production floor using Arena (www.arenasimulation.com) and Quest (quest.ucdavis.edu) simulation software, respectively. Simul8 DES software was used by Katsaliaki et al. [24] to create an HLA DS for a complex blood supply chain in the UK. They decided to turn to DS after being unable to conduct experiments with an OR/MS simulation of the whole supply chain in a desktop computer. They reported results on performance and concluded that for very large-scale simulation DS is a viable solution for efficient experimentation. Medina et al. [32] developed a DES DS for iron ore supply chain. They used Arena software to develop DES federates of import and export seaports. The communication between the simulation components and the RTI implementation was achieved via DLL functions developed in C++. They concluded that achieving interoperability and clock synchronization were the most challenging tasks. A distributed agent-based approach was taken by Long [30] for supply chain simulation. JADE multi-agent systems framework (jade.tilab.com) was used to model the agent components of the supply chain network. The system has its own coordination and synchronization implementation using agents for task regulations and time coordination. The author recognizes that some components of the proposed framework are still complicated and further experiments with real applications are needed. Tu et al. [51] used an upstream and downstream supply chain scenario of car manufacturing DS to demonstrate their approach for enterprise interoperability. They used HLA web services for model discovery and federation creation.

Significant standardisation efforts in the area of DES DS led to the international standard, by the Simulation Standards Interoperability Organisation (SISO), SISO-STD-006-2010. Taylor et al. [50] developed Interoperability Reference Models (IRMs) for linking models developed in commercial-off-the-shelf (COTS) simulation packages. Attempting to bring DS closer to OR/MS practices, this standard defined four IRM types (Type A. Entity Transfer, Type B. Shared Resources, Type C. Shared Event, and Type D. Shared Data Structure) describing interoperability issues when interfacing DES models. Type A describes the ways that an entity is transferred from one federate model to another. In order to capture the complexity of transferring entities in DES which consist essentially of queues and servers, three subtypes are identified (Type A1. General Entity Transfer, Type A2. Bounded Receiving Element, and Type A3. Multiple Input Prioritisation). The authors pointed out that IRMs can assist the development of DS where the federates have been created using DES commercial packages since they clarify communication issues between OR/MS modellers and DS experts. Also, the IRMs help in reducing the time needed for understanding and validating the way that models interact. The standard did not go further in proposing an OR/MS DS methodology.

Implementation of the above standard IRMs, and more specifically the Type A2 IRM, is presented by Strassburger et al. [47]. The authors developed a DS middleware for existing simulations of a tractor factory in South America. There were six existing simulations for different components of the production process. In line with the process, some component models were feeding the subsequent component models. Therefore, there is an entity transfer between models. The project aimed to investigate the sizing of the input buffers of the receiving components and the potential effects of blocking production sections due to full buffers. For this purpose, the connectivity requirements fell into the Type A2 IRM specifications. The authors reported that the implementation of the DS interface was straight forward after dealing with some time representation issues of the component models. Also, they pointed out the potential usefulness of the standardised IRMs in classifying and selecting the right solution for interoperability problems.

More work has been done in investigating the common problem for DS DES systems of entity transfer between two or more federates in industrial setting by Raab et al. [39] and Pedrielli et al. [38] who created DS environments using commercial simulation packages, and open source and commercial RTI implementation, respectively. Within this context, they investigated one IRM, the Type A.2 IRM of the SISO-STD-006-2010 [44]. The test case in Raab et al. [39] does not represent a real case study however it is an example of a typical production line that is used to draw interesting conclusions for the application, such as the complexity and the technical expertise required to implement the DS. Pedrielli et al. [38], on the other hand, used a sheet metal manufacturing case study. The problem in the latter case however was not the size of the simulation but rather the access to information for all interoperable components. DS allows subsystem models to interoperate using only information required for the interface while sensitive data can stay locally to the owner organisation. In this case, one of the participating factories would not share key data and therefore this lack of shared information made it impossible to develop an OR/MS simulation. Therefore, DS provided a convenient solution for modelling the information exchanged only at an interface level.

In terms of hybrid ABS-DES, where the goal is to interoperate ABS and DES simulations (for an introduction to ABS see Macal and North [31] and for DES see Robinson [42]), Mustafee et al. [33] identified the main connectivity approaches that currently exist. That is, to manually execute the models and transfer the relevant variables between them (non-synchronised manual execution), to automate the variable transfer but in a non-synchronised way (non-synchronised automated execution), and to use simulation packages that support multiple simulation techniques (synchronised CSP-driven execution) such as AnyLogic (www.anylogic.com). In their work in progress, Mustafee et al. [34] report on modelling offshore wind farms combining an ABS Netlogo (ccl.northwestern.edu/netlogo) model of the turbine degradation and a DES Simul8 (simul8.com) model of modelling maintenance, repair and operations strategies. The hybrid model required automated synchronous data exchange maintaining the causality order. Since the models were developed in different simulation packages, it was not possible to adopt the synchronised CSP-driven execution approach. The authors adopted the development of DS as a means of automated and synchronised approach for hybrid ABS-DES models execution. They identified the interactions between the ABS and DES models and developed the DS using HLA RTI. A DS environment in the context of offshore wind turbines construction safety verification was developed by Laesche et al. [26]. The authors proposed the simulation of machines and humans as agents and the physical environment as physical world simulation using the open source GameKit (github.com/gamekit-developers/gamekit).

Of the above work, most, if not all, presented a proof of concept or feasibility studies. The products do not go beyond the prototype stage. Most studies focus on technical issues, such as model interoperability, cross-platform interoperability and RTI performance rather than the real application. These discussions are beyond the limitations and concerns of OR/MS modellers, and the language sounds alien to their sector. All, more or less, describe the way that they developed their DS but none mention a systematic way for developing DS models. There is not a single recommendation in the literature that can be used as guidelines for conducting DS projects in the OR/MS world. Despite the DS promises of accommodating heterogeneity, data privacy, model autonomy, reusability, and computational resources little work has been done towards defining how these would be realised in OR/MS. In the next section, we attempt to bridge this "chasm" between the OR/MS and DS world views by mapping the corresponding activities in both OR/MS and DS practices for conducting a simulation project. This forms the basis for the proposed OR/MS DS methodological framework.

3. Towards an OR/MS DS methodological framework

The proposed OR/MS DS methodological framework is derived by combining well-established methods used as guidelines for conducting OR/MS and DS projects, respectively. There are many published methods for conducting an OR/MS simulation study. Their aim is to provide guidance to modellers for managing the various activities of simulation projects. This involves stages and distinctive steps to be followed, usually iteratively. Works from Ulgen et al. [53], Banks et al. [5], Law and Kelton [27], and Robinson [42] are among them. In OR/MS simulation methods, the steps in a project lifecycle are defined either descriptively or diagrammatically in the form of flow charts. Despite the different ways of representing the processes, generally they all agree that several steps are conducted in parallel, the whole exercise is highly iterative and that there are three main deliverables that should be produced. These are the *project plan*, the *model* including *conceptual* and *computer model*, and the *experimentation results*. The selection of the OR/MS simulation method to be used as basis for the OR/MS DS methodological framework was based on the criteria of popularity, adequate descriptive analysis of the simulation steps, and clear graphical representation that shows parallel activities, too. The one that met our criteria is proposed by Banks et al. [5]. The representation uses flowchart notation and a detailed descriptive transcript for each step accompanies the process flow. Furthermore, it is one of the most cited methods and clearly shows the parallel activities of model conceptualisation and data collection.

Efforts on providing guidelines for DS using the HLA led to the development of the IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP) standard (IEEE 1516.3-2003). FEDEP developed by SISO and standardised by the IEEE [19]. It recommends general practices for developing DS using HLA. These can then be tailored for the needs of individual applications. FEDEP is a superseded standard replaced by DSEEP which is a high-level framework aiming to provide guidance for developing three DS architectures and, as FEDEP, developed by SISO and standardised by the IEEE [18]. It facilitates the HLA, the Distributed Interactive Simulation (DIS) [20] and the Test and Training Enabling Architecture (TENA) (www.tena-sda.org/display). DSEEP is a generalised framework that can be adapted to meet individual DS applications needs. An example of an adaptation tailored to the requirements of specific organisational needs is the VEVA framework developed by the German Armed Forces, as mentioned in Siegfried et al. [43]. VEVA is a detailed engineering process aiming to facilitate the specific organisation. Both methods are developed by the defence sector and, although they can be used in OR/MS applications, they include practices familiar mainly to military simulation engineers. The systems engineering approach that both follow and the language used make them unattractive to OR/MS modellers. Since DSEEP is the only generic and active standardised process, it is selected as the basis for the DS part of our OR/MS DS methodological framework.

As established above, the two methods selected to denote the OR/MS and DS world views in our OR/MS DS methodological framework are the one described in Banks et al. [5] and the DSEEP, respectively. Initial comparison of the two methods led to a three-phase process. As in OR/MS, DS projects have the same three main deliverables. Therefore, the identified three phases correspond to these deliverables and are the *planning phase* where the *project plan* is produced, the *development phase* where the *model* is developed, and the *experimentation phase* where the *experimentation results* are delivered. In a top-level view, mapping the OR/MS and DS activities within the three phases led to the list shown in Table 1. The *planning phase* includes activities relevant to understanding and forming the problem as well as setting objectives and identifying resources for the simulation project. The *development phase* involves activities relevant to model(s) development, interfacing and testing. Finally, the *experimentation phase* involves activities relevant to simulation execution and result production and analysis.

Each of the top-level activities involves detailed activities that carried out in each phase. Aiming to understand the points of convergence and divergence between the OR/MS and DS practices, the following subsections will describe the corresponding detailed activities in each phase for both world views and highlight the functionalities for each. A mapping of these activities between the OR/MS and DS methods and within the three identified phases is shown in Fig. 2, where the pointing arrows show association. Some activities are associated with both OR/MS and DS and therefore there is convergence. Where there is divergence, the activities are associated only with the corresponding method.

Table 1
OR/MS and DS activities in planning, development and experimentation phases.

Phases	OR/MS activities	DS activities
Planning	• Problem formulation • Setting of objectives and overall project plan	• Define simulation environment objectives
Development	• Model conceptualisation • Data collection • Model translation • Verification • Validation	• Perform conceptual analysis • Design simulation environment • Develop simulation environment • Integrate and test simulation environment
Experimentation	• Experimental design • Production runs • Documentation and reporting	• Execute simulation • Analyse data and evaluate results

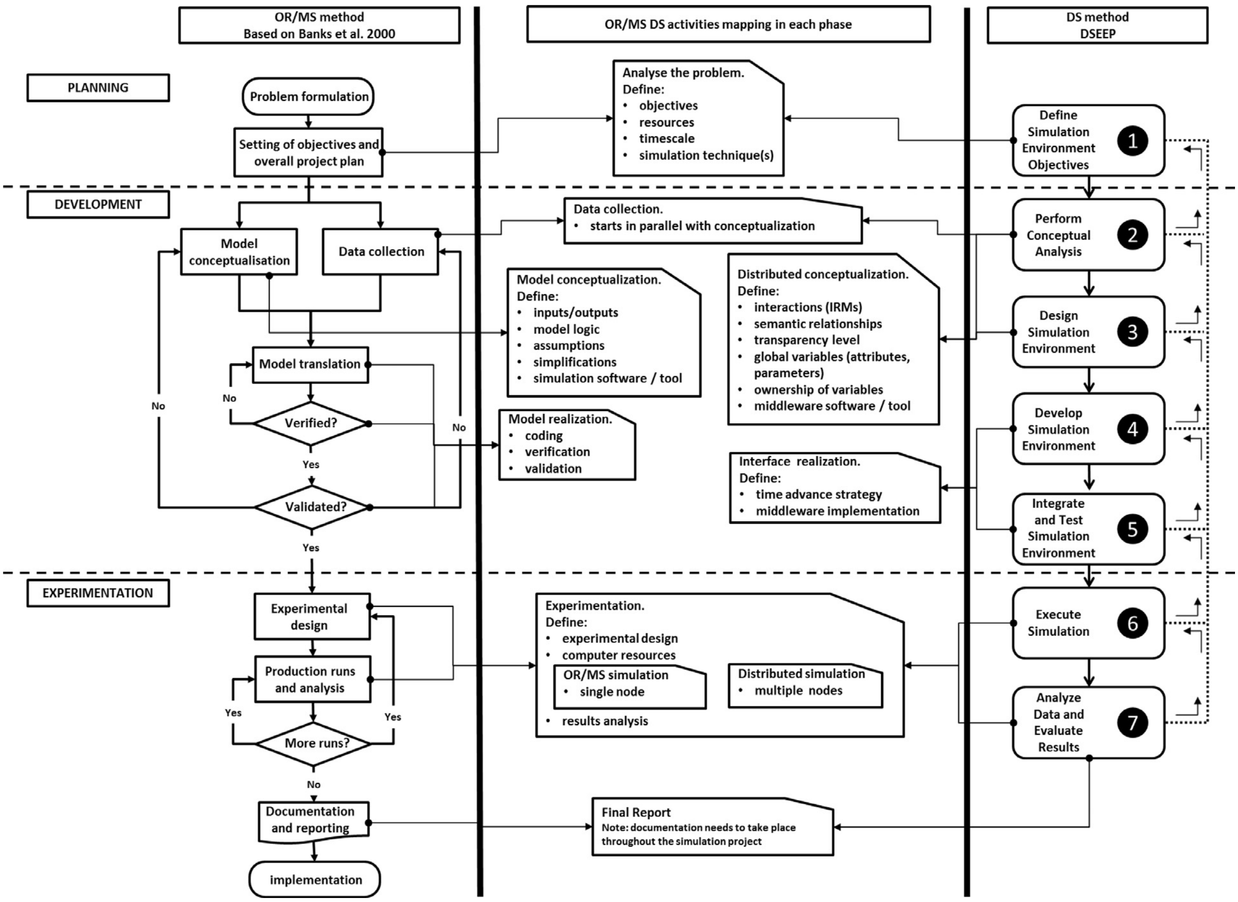

Fig. 2. Mapping of activities of OR/MS and DS methods in three phases.

3.1. Planning phase

The planning phase of a DS simulation is essentially the same as in OR/MS projects. During this phase the problem formulation takes place. That is, to understand the real world problem and determine the objectives of the study, also, the modelling team is formed and the time-scale and possible cost of the project are agreed upon. The most appropriate simulation technique is decided in this phase, too. Essentially, the main activity in this phase is to perform the simulation project planning.

3.2. Development phase

Three major activities can be distinguished in the development phase, that is, data collection, conceptualisation and realisation. The most fundamental differences lay in this second phase.

Data collection is an activity that takes place in a similar way in both OR/MS and DS projects. It involves high level contextual data for conceptualising the model and detailed data for developing the computer model [42].

Both conceptualisation and realisation should be dealt differently when developing OR/MS and DS projects. The conceptualisation of an OR/MS model includes activities such as definition of inputs and outputs, the model logic, assumptions and simplifications. However, in DS there is another consideration to be decided; what information is going to be exchanged between the component models. Additionally, the transparency of the federates, the global variables (e.g., attributes/parameters of the shared (published) objects/interactions) and the variables' ownership in the federation should be defined. Moreover, in a hybrid system where more than one simulation technique exists, there should be an understanding and documentation of the semantic relationships between the different techniques. For example, a fundamental element in DES is the "entity" which is an object that flows through the simulated process and its state changes by an activity. If this entity passes to an ABS in the DS it becomes an agent which is an object that can make decisions and change its behaviour according to some internal rules. It is critical to understand these relationships and map the related data for each element. Finally, the simulation software to be used should be decided in both OR/MS and DS projects. In DS projects, the RTI implementation to be used should be decided, too.

In the realisation of the simulation, the software that implements the conceptualisation should be developed. This can involve writing a program or creating the model using COTS packages. In both cases, the DS components should be developed too. In an HLA-based DS, apart from the simulation models, this actually means configuring the RTI components. Before implementing the RTI, however, the time management strategy should be defined. In HLA-based DS the time advancement is controlled centrally by the RTI; there are two time management services that are currently supported by HLA: Time Advance Request (TAR) and Next Event Request (NER). Each federate requests time advancement from the RTI, then the RTI responds with a Time Advance Grant (TAG) allowing the federation logical time to progress. In TAR, a federate requests time advancement to the next time step. However, when a federate is DES where a future event list is maintained, the time management service can be implemented as NER, where the federate requests logical time advancement at the time that the next event is scheduled to occur. Furthermore, the synchronisation protocol should be decided. A conservative synchronisation protocol ensures that the RTI will send messages with timestamp less that the current simulation time of a federate. This comes with a cost on synchronisation time overhead. An optimistic synchronisation protocol performs better in terms of latency. The optimistic time management federation allows violation of the local causality constraints, however, the optimistic algorithms provide error recovery mechanisms in the case that a causality error has occurred. Therefore, optimistic synchronisation supports rolling back in simulation time and recovering system states prior to the error event.

In both OR/MS and DS projects, the code must be verified and the simulations must be validated. In OR/MS simulation, model validation usually involves comparing the outputs with a real system or expected outcomes. However, in DS, the causality and the whole system behaviour should be tested too. Validation against federation synchronisation requirements and the proper implementation of the IRMs should be conducted.

3.3. Experimentation phase

DS is normally executed over a network. Therefore, the experimental design should provide for this feature. The availability of network resources of distributed computing infrastructures should be provisioned and the connection requirements, such as bandwidth and security, should be decided. Apart from the computer resources, the experimental design and the analysis of the results are the same in both OR/MS and DS M&S projects.

4. Proposed OR/MS DS methodological framework

The ultimate goal is to merge the DSEEP activities into the OR/MS simulation lifecycle in a way that is agreeable with the OR/MS practises. The proposed methodological framework is derived by combining aspects of OR/MS and DS methods following the above discussion. Taking into account the necessary activities from DSEEP and modifying slightly the OR/MS simulation method by Banks et al. [5], our OR/MS DS methodological framework is illustrated in Fig. 3.

It is highlighted that the documentation process commences from the first step and continues throughout the simulation project. This documentation leads to the final report of the project. The small alterations in the OR/MS method by Banks et al. [5] are the two parallel activities. Namely, the data collection and the documentation processes. From experience, the computer simulation program can be verified even when the model is not populated with the actual data. The model coding, as well as the data collection, process can be very time-consuming. By performing the two activities in parallel fashion, valuable time can be saved. Furthermore, as mentioned by Onggo and Hill [36], the two activities are closely interwoven. For example, the data availability, or rather unavailability, and quality can lead to model alterations. Similarly, the parallel documentation of each step can help avoiding omissions in the final report. The activities for an OR/MS DS project in each phase are described below.

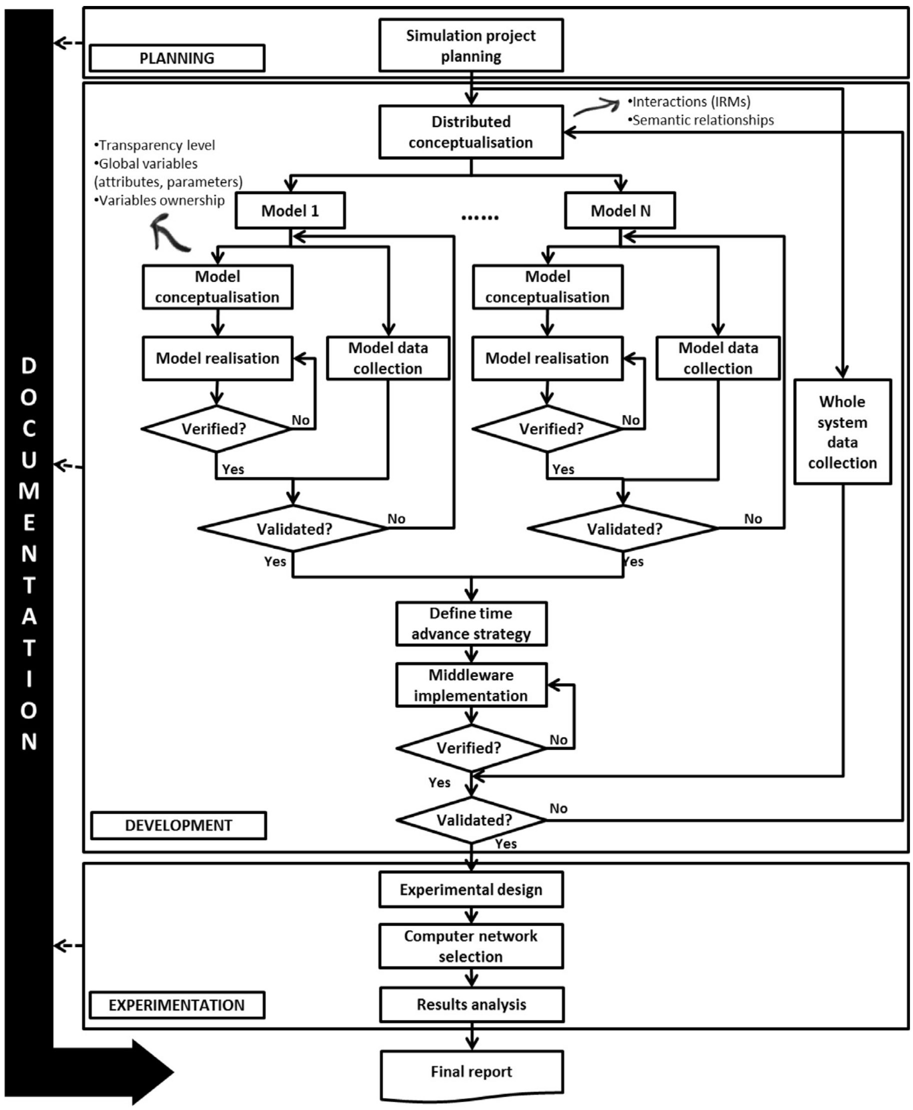

Fig. 3. OR/MS DS methodological framework.

4.1. Planning phase

This first phase is where the simulation project is shaped. The main deliverable is the simulation project plan. Considering the activities in both OR/MS and DS methods, the main activity is the simulation project planning and involves the tasks explained below.

4.1.1. Simulation project planning

In this step, the problem to be analysed should be defined first. This involves all stakeholders and it must be ensured that all involved parties, e.g., managers/policy-makers and modellers, achieve a common understanding and agree of the problem to be simulated, and indeed whether simulation is appropriate for problem analysis. Then, the project objectives should be set and the questions to be answered by the simulation study should be specified, and the key performance indicators (KPIs), i.e., the performance measures to be output by the simulation project, should be determined. Lastly, the required resources, the timeframe and the plan of actions and milestones are identified. In this stage, the appropriate simulation technique(s) is decided. Here, the project team should decide whether DS can be employed, as well as whether existing models will be reused.

4.2. Development phase

4.2.1. Distributed conceptualisation

In the distributed conceptualisation step, the conceptual model of the whole system should be delivered. This involves the component models (federates) and their interactions. At this stage, it is not necessary to include the details of each model. The important elements are the subsystems that each model will represent and the communication among them. The individual models therefore can be represented as black boxes. The interface however between each of the interoperating models must be defined. In other words, IRMs should be defined in order to describe what and in which way the models communicate. The IRMs include the exchanged information but does not necessarily include time synchronisation information. The time dimension is only mentioned when there may be a conflict and the order of events must be in the specified relationship. It should be mentioned here that the standard IRMs (SISO-STD-006-2010) that had been developed considering interoperability requirements for DES can be easily transferred to other discrete time simulations, e.g., ABS. However, if the DS is hybrid, i.e., more than one simulation techniques is involved, the semantic relationships between the different techniques must be identified. This is necessary for the common understanding of the information exchange. Usually a DS model involves more than one standard IRM.

4.2.2. Whole system data collection

This data involves detailed information about the whole system. The activity of collecting the data starts in parallel with the distributed conceptualisation activity. This data will be used in designing and validating the federation.

4.2.3. Model/federate building

Building models that can be used as federates in HLA does not differ radically from the model building process of an OR/MS simulation. Therefore, in this Section 4.2.3 we will explain only the additional actions required for a DS model. Furthermore, if the federate models exist, they can be reused with some modifications for including the HLA components.

4.2.3.1. Model conceptualisation. An additional requirement in OR/MS model conceptualisation activity, when developing the model as federate, is to define the transparency level. That is, to denote what internal information can be seen by the whole federation. This activity will be guided from the distributed conceptualisation and the IRMs. Then the global variables should be decided. The global variables values will hold the values to be updated for sending to, or receiving from other federates via the RTI. Finally, the ownership of the variables should be decided. That is which federate can update the values.

4.2.3.2. Model data collection. This activity involves collecting data for modelling each subsystem. It starts in parallel with conceptualising the subsystem model. If the model is reused, this activity may be omitted if the data has not been updated since the previous model building.

4.2.3.3. Model realisation. This activity involves the simulation coding, verification of the produced code and the validation of the subsystem model, exactly as in OR/MS modelling.

4.2.4. Define time advance strategy

As mentioned in Section 3.2, in HLA-based DS there are two time management services. Depending on the design of each federate and the simulation technique(s) that is modelled, TAR or NER can be selected. Also, the synchronisation protocol should be decided here. The synchronisation protocol should reflect the causality maintenance requirements of the DS system. As said above in Section 3.2, the synchronisation protocol can be conservative or optimistic.

4.2.5. Middleware implementation

This is actually the coding of the RTI implementation. The absolutely necessary components that must be developed are the RTI ambassador, the federate ambassador, and the Federation Object Model (FOM) and Simulation Object Model (SOM). The RTI ambassador is responsible for the receiving messages from the RTI, the federate ambassador is responsible for sending messages to other federates through RTI. The information exchanged during runtime is defined in the FOM. FOM is common for all federates in the federation and specifies what objects and their attributes and what interactions and their parameters are exchanged via the RTI. In HLA objects are fundamental elements that their characteristics (attributes) may be published by a federate and other federates may subscribe to. Interactions are events that may affect other federates and therefore their parameters may be published by a federate and other federates may subscribe to. Each federate has its own SOM that defines the information exchange in the federate level. If a SOM is compatible with a FOM, then the federate can join the federation. Verification involves testing the middleware code. The validation of the DS happens after all component models and the middleware are complete in order to validate the whole system. This involves testing the time synchronisation and the causality of the whole OR/MS DS model.

4.3. Experimentation phase

The experimentation phase involves activities for designing and running the experiments with the simulation model, as well as analysing the outputs and producing results.

4.3.1. Experimental design

Experimental design for an OR/MS DS includes decisions similar to any OR/MS simulation experiment. The modeller here decides whether initialisation or warm-up period is needed and if so, how long this period will be. Also, the length and the number of simulation runs using different random numbers are decided. In addition, the experimental design involves defining the scenarios to experiment with, including the identification of the input parameters and the range of their values in the defined scenarios. Further details on OR/MS experimental design can be found in Bank et al. [5], Law and Kelton [27] and Robinson [42].

4.3.2. Computer network selection

While OR/MS simulations run typically on a single desktop computer, DS runs on a computer network. The type of the network needs to be decided regarding the project requirements. Among the considered requirements are location, privacy and availability of computing resources.

4.3.3. Results analysis

Results analysis in OR/MS simulations involves numerical and graphical analysis of the KPIs, sensitivity analysis and solution recommendation [42]. When the system is designed as DS, insight in the subsystems interdependencies can be gained. The results analysis can include the impact one subsystem's performance has on another subsystem or on the whole system.

To illustrate our OR/MS DS methodological framework processes, we give a detailed walk-through of an example of a hybrid ABS-DES DS of a large Emergency Medical Services (EMS) system based on pervious work (see [2] and [3]).

5. The London emergency medical services case study

This section describes the steps of the framework using a practical example of a simulation of a large and complex system within the healthcare sector.

5.1. London EMS planning phase

5.1.1. Simulation project planning

Understanding the real world problem. Generally, healthcare organisations are complex, large systems and most frequently with heterogeneous dynamics. Many endogenous factors, across multiple healthcare organisations, affect the performance of the whole system. For example, the performance of an ambulance service is grossly impacted by the operations of the regional emergency departments, and these in turn are affected by delayed hospital discharge, etc. in a cascading fashion. Moreover, exogenous factors such as the politico-economic climate, the growth and ageing of population affect the resources and pressure on the healthcare system. In the current economic status and the continuous growth and aging of populations, the healthcare sector faces the challenge of delivering high quality services with fewer resources to larger populations. Efficient management and forward planning, not only locally but rather across the whole system, could support healthcare sector to overcome the challenges. Ambulance services and Accident & Emergency (A&E) departments are closely interwoven organisations in an EMS system. A comprehensive analysis should include both. Nonetheless, the analysis of such large and complex systems is massively difficult without the proper decision-making and system analysis tools.

Establishing whether M&S is appropriate for problem analysis. One such tool is M&S, which, with the advancements in computing, has presented considerable theoretical and technological progress. M&S is being used in a range of scientific disciplines to study new systems or changes to existing ones and answer "what-if" questions when the physical implementation of a system is difficult or even impossible to be achieved. Naturally, simulation uses historical data as input and outputs an estimation of the system behaviour considering the randomness of the system variables. Arguably, a well-designed simulation can reach sufficient accuracy. The quality and validity of the outputs are highly dependent on the quality of input data and the replications of the experiments. Sometimes the experimentation size is compromised by the available computing resources. Contemporary computing infrastructures that offer large storage and memory capacity, and multiple processors in affordable costs provide a new perspective in the area of M&S. Therefore, M&S that can be executed in a distributed infrastructure is considered desirable tools for better informed decision making in the scale of EMS systems.

Define the objectives of the M&S study, determine KPIs and identify resources. Having determined that M&S is the suitable approach for the analysis of such a complex system as EMS, the aim of the study was to investigate novel approaches for EMS analysis using discrete time M&S and the main objective was to demonstrate the feasibility of these approaches. The KPIs for the whole system were performance measures such as execution time and scalability. Also, the modelling team had the expertise and time to complete the project. Therefore, an analysis of the discrete time M&S EMS studies recently published in the academic literature followed in order to identify the gap in the area.

There are many articles that present simulation studies of the ambulance services. These studies focus on a variety of problems that ambulance services policy-makers face, such as dispatching strategies, scheduling, covering, etc. For example, improvement scenarios of the French ambulance service were studied by Aboueljinane et al. [1] using DES. They developed their model using the commercial simulation software Arena. Ibri et al. [17] modelled the emergency medical services in Switzerland using MAS. Air-ambulance services were modelled by Lee et al. [28] using Korean trauma cases data. The

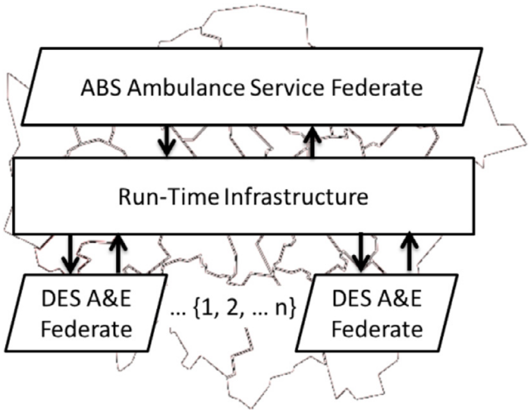

Fig. 4. Hybrid ABS-DES DS for London EMS.

authors used DES with mathematical optimisation modelling developed in C#. Van Buuren et al. [54] developed an EMS simulator for the Netherland's ambulance service in the Amsterdam region. For portability purposes, they used C++ programming language to realise the simulator. It is a discrete time simulation however it can run in a speed mode as DES, where an ordered list of current and future events is maintained. At the same time, a lot of simulation studies focus on the operations of the A&E departments. For instance, Wang [56] presented a prototype ABS model, developed in Netlogo, of the emergency department at the University hospital, Virginia. The author argues that ABS may be adopted easier than complex mathematical models by healthcare managers. DES using the commercial simulation software FlexSim Healthcare (www.flexsim.com/flexsim-healthcare) was utilised by Holm and Dahl [16] for modelling the Akershus University Hospital emergency department Norway. The model was developed to study the effect of an expected increase in patient volume. A DES model of a busy A&E department in West London, UK was developed by Eatock et al. [10] using the commercial package Simul8 (www.simul8.com). Paul and Lin [37] used DES to investigate the reasons for patient overcrowding in emergency departments and identify strategies for resolving them. They developed the model using ProModel (www.promodel.com) simulation software. The overcrowding of the emergency departments was studied by Ashour and Okudan-Kremer [4], too. The authors developed a DES model using the commercial package Simio (www.simio.com). Lim et al. [29] modelled the clinical staff of an emergency department as pseudoagents (i.e., entities with embedded decision logic) in a DES model using Arena software. Instead of the conventional resources of the DES technique, they introduced interactions among the clinical staff. Rahmat et al. [41] used ABS to model the re-triage process within emergency departments. ABS was selected in order to model the interactions among the emergency department objects. An ABS model was developed by Wang et al. [55] to study preparedness and response of an emergency service to a mass casualty event in urban area. The authors developed their model in Repast Simphony (repast.sourceforge.net). The topology of the ABS model is a GIS that provides data related to road networks and location of the ambulance vehicles and hospitals.

Two obvious conclusions were drawn from the literature; first, many studies discuss models developed for ambulance services and A&E departments for a specific purpose and there is no plan for model reusability; second, there is lack of models that incorporate all the components of an EMS as a whole system. Most EMS simulations just acknowledge the A&E congestion. However, little evidence from the literature indicates that the whole EMS system and the interactions among the different subsystems have been adequately studied. This is due to multiple reasons. First, developing all the involved models requires massive effort from modellers, second, the required data may not be available, third, the execution of such large and complex simulations are very computationally expensive. Therefore, DS could be really beneficial for EMS simulation, where reuse of existing simulation models that run on different nodes of a computer network and can be composed to form a large distributed simulation model is supported.

Determine the appropriate M&S techniques. Two are the main M&S techniques that are commonly used to analyse healthcare in the organisational context: DES, and ABS. DES and ABS are both microscopic simulation techniques that are characterised by high level of details. The time advancement is discrete. Usually, in DES the time steps are not fixed but there are time advances to the simulation time that the next event is scheduled to happen. DES engines keep a future event list for this purpose. ABS time progresses in fixed time steps. Both DES and ABS can model a system at a low level of individual objects. DES entities are objects that go through a process and their state changes as the simulation time evolves. ABS agents are objects that "live" in an environment and interact with it and with each other. Agents have the ability to learn and change their behaviour. Due to the process-oriented character of A&Es, DES was selected for the hospital simulation. On the other hand, the ambulance service model can benefit from the autonomy of the agents and their ability to interact with the environment and adjust their behaviour; therefore ABS was selected for the ambulance service. A high level diagrammatic representation of the federation is shown in Fig. 4. DES and ABS are highly stochastic techniques where several replications are executed to increase the confidence of the simulation outputs. Due to unavailability of existing models, all federates were developed anew.

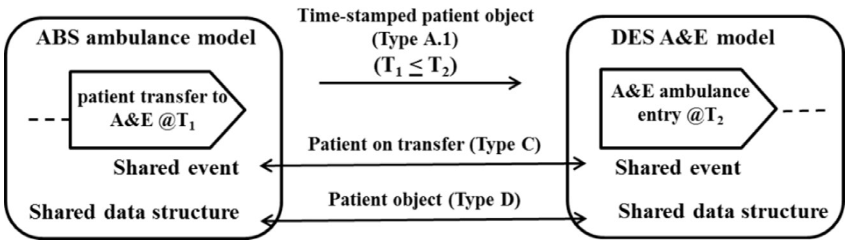

Fig. 5. EMS IRM.

5.2. London EMS development phase

5.2.1. Distributed conceptualisation

In the London EMS hybrid model, two simulation techniques are involved, these are ABS and DES. Between ABS and DES there are semantic differences and therefore the semantic relationship was defined. For example, an agent in ABS can be either entity or resource in DES. A queue in DES is not modelled explicitly in ABS. Importantly, DES is event-driven while ABS is time-driven simulation. Regarding the IRMs that can define the interoperability between the component models, in the EMS, three types of interchanged information can be identified. First, all A&E department models should be able to communicate their availability to the ambulance model. It is essential for the functionality of the system, that when the ambulance model searches for the most appropriate hospital to transfer a patient, the most up-to-date information about all A&E departments' availability is known to the ambulance service. Second, the patient object/agent should be transferred from the ambulance service model to the ambulance patient object/entity entry point of the A&E department model. Third, the ambulance service model should be able to notify the A&E department model that a patient is on transfer in order to reserve resources and avoid conflict in the hospital availability level. Furthermore, the patient object and all its properties must be known to the A&E model. Therefore, in this project, the interactions between the two parts of the hybrid model can be represented by a tuple Type(A.1, C, D) or Type(general entity transfer, shared event, shared data structure) of the SISO-006-2010 standard, as shown in Fig. 5. There is a time restriction that must be defined in the A.1 IRM. That is, the time that a patient leaves the ABS LAS model must always be less or equal to the time the same patient arrives at one of the DES A&E models. The software tools selected for this project were open source, both for the model development and the RTI implementation. As a simulator, the Repast Simphony (RepastS) toolkit was used. RepastS is essentially an ABS toolkit however it can easily be converted to DES simulator since both ABS and DES are discrete time simulations. To do that, the fundamental components of a DES, namely queues, work stations and resources, were hard-coded. For the interface, the Portico v2.0 RTI implementation of IEEE-1516-2000 (www.porticoproject.org) was used. Finally, a simplification due to time constraints was necessary and therefore it was decided that only general A&Es in the Greater London area would be included.

5.2.2. Whole system data collection

All data in this project was collected from publicly available online NHS England archives for the year 2011–12. The data collection process can be separated into two parts. First, data is needed for the whole federation. In our example, we need data on the entire London EMS. That is the number of A&Es in the area and their exact location, as well as the type of services they provide e.g., if they are general or specialised A&Es and if they have paediatric units. In this implementation we included general A&E departments in the area of Greater London. At the period of data collection there are 32 general A&Es in the coverage area of the London Ambulance Service (LAS) which covers an area of 620 square miles. The second part involves data for each A&E and the LAS and this belongs to the model data collection activity which will be discussed in the next subsection.

5.2.3. Model/federate building

In the EMS system, we have two organisations the hospital A&Es and the ambulance service. In respect to the project objectives, the level of detail for the federate models is decided here. In this example and because the aim of the project was to investigate novel approaches for EMS analysis with the objective to demonstrate the feasibility of them, both systems where modelled with low level of details. An assumption was made that the A&E always has available resources when an ambulance patient arrives therefore the handover time was not modelled.

General A&E: As in any OR/MS DES simulation, the processes, activities and resources were identified. As mentioned earlier, patient arrival data was collected from online resources, while data on activity times and resources was estimated using previous experience in A&E modelling [10]. KPIs were the waiting times, as this is a performance measure for A&Es. The difference in the design in order to support individual execution as well as interoperate in the DS was in the ambulance arrivals. In an OR/MS simulation execution, patients arrive using the appropriate distribution, while when running as a federate, ambulance arrivals are received from the ABS model. Each A&E exchanges information only with the LAS model.

Table 2
A&E model input data.

A&E model input data			
Walk-in inter-arrival time normal distribution		**Patient condition**	
Mean	4.81	Minors	35%
SD	0.59	Majors	65%
Time in triage normal distribution (with staff)		**Need treatment**	
Mean	7.00	Yes	60%
SD	2.00	No	40%
Time in minors normal distribution (with staff)			
Mean	30.00	**Number of staff**	15
SD	10.00	**Triage capacity**	5
Time in majors normal distribution (with staff)		**Minors capacity**	12
Mean	40.00	**Majors capacity**	24
SD	10.00		

Table 3
Ambulance model input data.

Ambulance model input data					
Inter-arrival time normal distribution		**Patient condition**		**Average speed**	15 mph
Mean	2.52	Minors	26%	**Correction factor**	1.32
SD	0.09	Majors	74%	**Coverage area**	320 sq
Time on scene (min) normal distribution		**Need transfer to A&E**		**Ambulance stations**	35
mean	22.52	yes	62%	**Ambulances**	187
SD	10.54	no	38%	**Hospitals**	16

Hospitals are not aware of each other. All internal A&E processes are hidden from the ambulance service model apart from its availability, resources and ambulance arrivals. The input data for the A&E departments DES models is shown in Table 2.

London Ambulance Service: For ABS simulations, the OR/MS simulation development followed the standard procedure for ABS modelling. The environment and topology were decided, the agents and their interactions were defined. The hospitals are present in the ambulance model as part of the environment. Once a patient is transferred to an A&E; when executing as an OR/MS simulation there is a treatment time sampled from a distribution, then the patient is discharged and the hospital availability is calculated locally; when distributed the patient passes to the A&E DES model and is removed from the ABS model. In the same way as in the A&E models, hospital federates are not aware of the internal logic of the ambulance model apart from the patient that is on transfer and its properties. LAS, at the data collection period, have 70 ambulance stations and 375 ambulance vehicles. Other types of vehicles, such as two-wheel vehicles and fast response cars, are not included in the model. The average speed in London is relatively low ranging from less than 10mph in central London and up to 22mph in greater London [45]. In the model an average speed of 15 mph is considered. Also, the distance in urban setting can be calculated with a high degree of confidence using Euclidean distance with a corrective factor [23]. The experiments were conducted using a federation half the size of LAS, approximately. The reason for this scaling down was that it was not possible to run the whole federation in a single node in order to test the performance of distributed against single node execution. The input data for the ambulance model is shown in Table 3.

For all federates the verification process was done iteratively with the model programming. The A&E waiting time was used to validate the model against the real system and the ambulance response time was used to validate the ambulance model.

5.2.4. Define time advance strategy

As mentioned above, this is a hybrid federation that consists of DES and ABS models. In ABS, as a time-driven technique, there is no event list maintained. Based on that, the time advanced strategy that suits the specific federation is implementing a TAR time management service. A conservative synchronisation protocol was implemented in this study.

5.2.5. Middleware implementation

In this example, each A&E federate has a unique identifier which corresponds to the A&Es in the ABS environment. The ambulance model's published attributes are the ID of the selected A&E, the condition of the patient on transfer, and the time that the ambulance with the patient arrives at the chosen A&E. The A&E models subscribe to the above attributes so as to receive the information. In turn, the A&E models publish their availability and the ambulance model subscribes to this attribute. All this information is defined in the FOM XML document. In this particular implementation, SOMs were not defined. Portico is a fully decentralised implementation. The only requirement is for each federate to be able to access the Portico library and to implement the RTI module that includes the RTI Ambassador and the Federate Ambassador classes. The RTI Ambassador is responsible for sending updates (published attributes/parameters) and the Federate Ambassador for receiving updates (subscribed attributes/parameters) via the RTI. To achieve integration, instances of the *RTIAmbassador* and *FederateAmbassador* classes are added in the RepastS context when initialising the federate (using the RepastS *ContextBuilder* Interface). For publishing the updates in the earlier defined attributes, the *updateAttributeValues()* method of the *RTIAmbassador* class is used. Before doing the actual updates, we declared the handle variables that keep the values of the attributes that will be communicated and added them to the attribute *Collection*. The update method in the *RTIAmbassador* for the Ambulance federate is shown below:

```
public void updateAttributeValues(int hID, double arrivalTime, int pCondition)
throws RTIexception
 {
   AttributeHandleValueMap attributes = rtiamb.getAttributeHandleValueMapFactory().
   create(3);
   HLAinteger32BE aaValue = encoderFactory.createHLAinteger32BE(hID);
   HLAinteger32BE abValue = encoderFactory.createHLAinteger32BE(pCondition);
   HLAfloat64BE aeValue = encoderFactory.createHLAfloat64BE(arrivalTime);

   attributes.put(aaHandle, aaValue.toByteArray());
   attributes.put(abHandle, abValue.toByteArray());
   attributes.put(aeHandle, aeValue.toByteArray());

   HLAfloat64Time time = timeFactory.makeTime(fedamb.federateTime+fedamb.
   federateLookahead);
   rtiamb.updateAttributeValues(objectHandle, attributes, generateTag(), time);
 }
```

Furthermore, the *FederateAmbassador* is responsible for receiving the updates of the variables that we have subscribed and decoding them. This is implemented in the *reflectAttributeValues()* method. For example, the Ambulance federate subscribes to the A&E availability (two different values for the minors and majors departments). Also the A&E federate sends its identification. This is implemented in the *FederateAmbassador* as follows:

```
for(AttributeHandle attributeHandle: theAttributes.keySet())
  {
    if(attributeHandle.equals(federate.miHandle))
    {minorHosAvailability=decodeInt(theAttributes.get(attributeHandle));}
    if(attributeHandle.equals(federate.maHandle))
    {majorHosAvailability=decodeInt(theAttributes.get(attributeHandle));}
    if(attributeHandle.equals(federate.idHandle))
    {HospitalID=decodeInt(theAttributes.get(attributeHandle));}
  }
```

The request for time advancement is implemented in the *RTIAmbassabor* class using the *advanceTime()* method. This method is annotated as *@ScheduleMethod()* and therefore is added to RepastS scheduler.

The middleware verification process was done iteratively with the RTI module programming. The validation of the middleware was done by corresponding the A&E unique identifiers in the ABS environment and the chosen hospital federate, the ambulance patient attributes, and the arrival time to A&E in both ABS and chosen DES models.

5.3. London EMS experimentation phase

5.3.1. Experimental design

As defined in the planning phase, the main objective of this study was to demonstrate the feasibility of the DS for EMS M&S approach. Therefore, the experimental design included performance and scalability testing. For performance testing, the conducted experiments involved two-dimensional increase in the simulation size; first, by increasing the number of federates in the federation, and second, by increasing the simulation time. For scalability testing, the experiments involved increases in the number the objects (emergency calls arrivals and A&E walk-in arrivals) and consequently events in the federation.

5.3.2. Computer network selection

The federation execution performed in a homogeneous non-dedicated network interconnected via LAN connected with 1 Gbps network card. Each node had an i5-2500 processor at 3.30 GHz speed and 4.00 GB RAM running Microsoft Windows 7 with Java 1.7 JRE and Portico v2.0 installed.

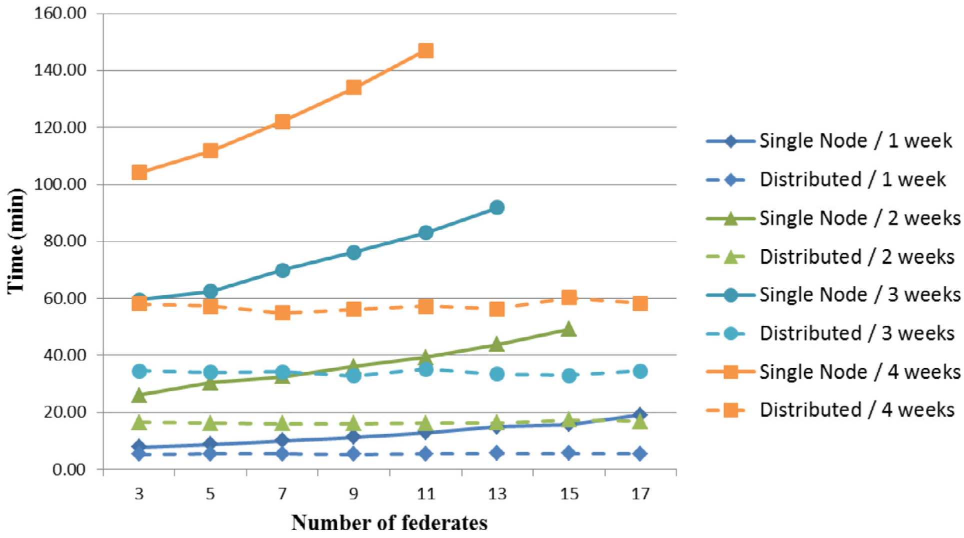

Fig. 6. Execution time (number of weeks run) vs number of federates.

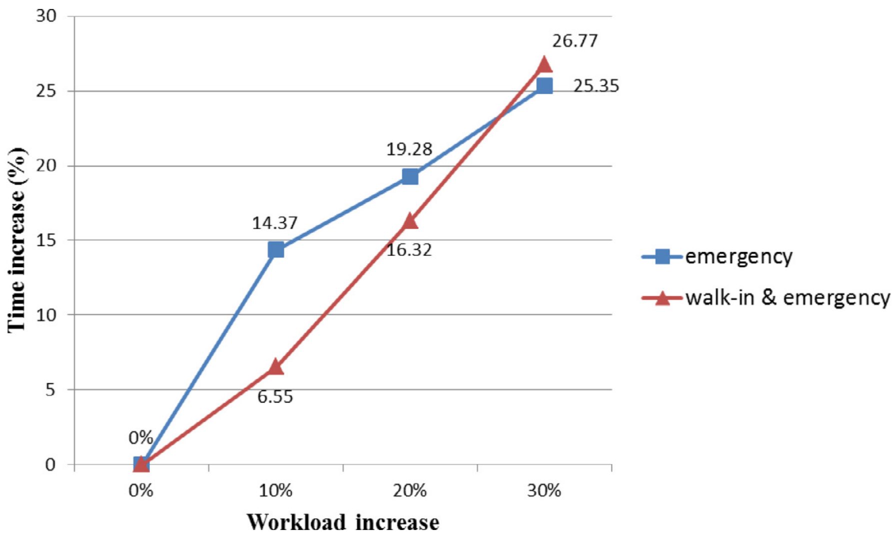

Fig. 7. Comparative scalability of increased workloads.

5.3.3. Results analysis

The results analysis involved graphical representation of performance and scalability outputs and calculation of speedup, which are illustrated in Figs. 6–8 respectively. Fig. 6 shows the execution time versus the number of federates. As can be seen, the equivalent single simulation is limited by size/execution time (four weeks limits the single simulation to the equivalent of 11 federates).The distributed version always performs better, with the best execution difference being the 4-week run. The single simulation gradually increases in execution time with extra simulations (until the capability of the single node is reached). The distributed simulation execution time remains roughly the same. This reflects the interactions between the A&E simulations and the ambulance – at no time does the ambulance federate become a bottleneck and the A&E federates are able to process their workload in parallel. Fig. 7 shows workload scalability. It is realistic to investigate sensitivity in A&E systems by investigating the impact of extra patients. This graph shows the impact on execution time from adding between 10–30% extra loading from extra emergency and walk-in patients. As can be seen, the distributed simulation execution time roughly reflects extra simulation work due to extra events that need to be processed in each of the simulations due to the total increase in patients (i.e., each A&E takes longer to process its workload). Fig. 8 shows the speedup of the distributed simulation compared to a single simulation for one week run. The trend reflects the increase in runtime for the single simulation against the relatively constant execution time of the distributed simulation.

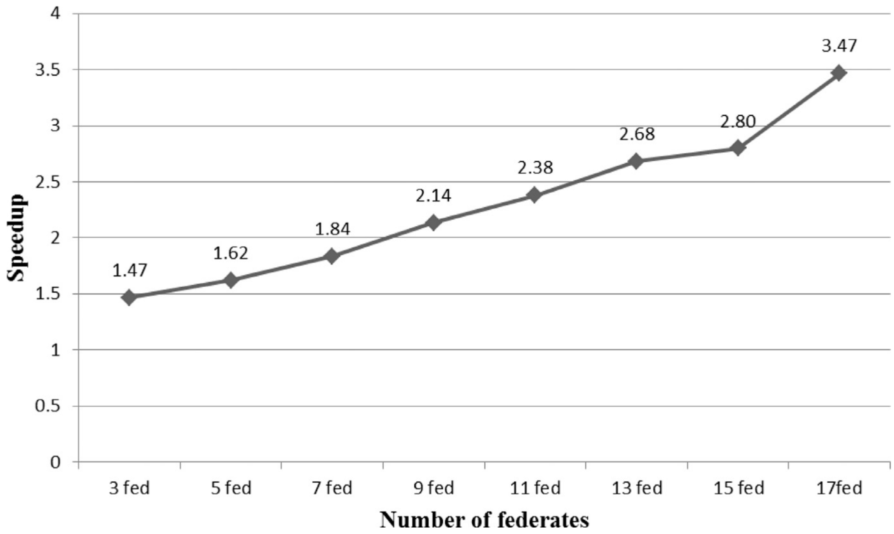

Fig. 8. Distributed simulation speedup relative to single simulation (1 week run).

Table 4
OR/MS DS methodological framework application.

Phases	OR/MS DS activities	London EMS example
Planning	• Simulation project planning	For a comprehensive analysis of an EMS system, both ambulance services and hospitals and their interactions should be included in the analysis. M&S was established as an appropriate technique for studying EMS and the objectives, as well as the KPIs, were defined as demonstrating the feasibility of new M&S approaches, such as DS, for such systems. ABS for the ambulance service and DES for the hospitals were selected as the appropriate M&S techniques for the individual models.
Development	• Distributed conceptualisation	ABS-DES semantic relationships. Identify interactions between federates, i.e., define IRMs.
	• Whole system data collection	Identify the number, type and location of hospitals in the LAS coverage area.
	• Model / federate building • Model conceptualisation • Model data collection • Model realisation	The three activities of model conceptualisation, data collection and realisation are under *model / federation building,* since these are the usual OR/MS model building activities. The only difference lies in the model conceptualisation activity where the transparency level was defined, i.e., it was decided that the ambulance model will share information only about the patient on transfer to a hospital with the selected hospital, all hospital models will share their availability with the ambulance model. The ownership of the global variables (attributes) will stay locally to the originator, e.g., the ambulance model does not have the privilege to modify the availability of any hospital. No interactions were defined in this implementation.
	• Define time advance strategy	A TAR time management service and a conservative synchronisation protocol were implemented.
	• Middleware implementation	Implemented RTI and federate ambassadors and FOM.
Experimentation	• Experimental design	Performance testing by increasing the number of federates and the simulation time. Scalability testing by increasing the objects (agents/entities) of the system.
	• Computer network selection	A LAN with 1 Gbps network card was utilised.
	• Results analysis	Performance and scalability analysis.

6. Discussion

The previous section presented the development of the London EMS DS using the OR/MS DS methodological framework proposed in this paper. The framework attempts to bring together practices used by two simulation communities. On the one hand, there are the OR/MS simulation experts that use simulation packages to develop models with some modular programming. On the other hand, there are the federated distributed simulation experts that use software engineering practices to develop simulations and the HLA standard and associated software to implement communication between different simulations. The activities carried out in each step and their reasoning explained in detail in the context of the application. For a comprehensive view, a summary of all activities under each phase is outlined in Table 4.

The proposed OR/MS DS methodological framework provided a clear guidance for building the London EMS hybrid DS model. Organising the required activities within the usual OR/MS practices as well as explaining them in a non-software engineering terminology was significant help in the whole management of the project. Yet, we acknowledge the complexity of developing DS compared to OR/MS models, especially when modellers are used to create their models using COTS that do not require programming knowledge. For DS development, there is certainly a need to program the middleware and mastering the HLA technical details presents a steep learning curve. The OR/MS DS methodological framework however simplified the process and helped to appreciate that the differences between OR/MS and DS modelling are not dramatic. Importantly, it contributed considerably to conceptualisation of the DS model.

Arguably, shifting the way of thinking from the OR/MS modelling to DS is the most challenging task. In our implementation, distributed conceptualisation was the most demanding and prolonged activity. In this example, identifying the submodels was obvious enough since each submodel represented an individual organisation. But there are systems that this is more complicated, especially when the whole system sits within a single organisation and even a single department. However, identifying the interactions and defining the IRMs, was quite challenging. For example, we decided to model the patient transfer as a general entity transfer (Type A1). This was achieved by enquiring the hospitals' availability prior to selecting the appropriate A&E. However, it could have been modelled as bounded receiving element (Type A2) where the hospitals could have blocked their ambulance admissions if they were crowded. This would be a good implementation if, for example, ambulance diversion was the focus of the study. It could also have been implemented as multiple input prioritization (Type A3), if we allowed multiple ambulance patients to arrive at the same A&E at the same simulation time.

The above decisions are critical in a DS implementation and careful consideration should be taken into distributed conceptualisation. Usually, solutions to such issues in OR/MS simulations are implemented by the COTS simulation packages. For OR/MS modellers this is a shift in the way they develop their model. Moreover, in this example we developed a DS using only the minimum required HLA functionalities. HLA is a complicated standard that covers comprehensively all (or nearly all) possible services for DS. OR/MS modellers usually need to implement only a fraction of these. Arguably, presenting the whole process in a familiar and simplified method to guide the project development is a considerable step in bringing together OR/MS and DS approaches.

Having had used DSEEP, the development of the London EMS DS would be possible but considerably more laborious. DS practices are not familiar to OR/MS modellers. The DSEEP recommended practice that presents detailed guidelines for DS uses a language that is not familiar to OR/MS. Therefore, it would be necessary first to understand the whole DSEEP recommended practice and then decide what is necessary for our project. On the other hand, if we wanted to use OR/MS methods, the DS implementation would be totally unguided since none of the OR/MS methods includes DS components.

A method that combines OR/MS and DS practices is missing from the simulation literature. Our OR/MS DS methodological framework is an additional step towards building theoretical tools for DS for OR/MS applications. Supported by practical example, it provides a guide for OR/MS DS projects that merges the DS practices, described in DSEEP, and the OR/MS practices presented in the popular OR/MS method by Banks et al. [5]. It may be argued that well-developed theoretical frameworks for DS in the context of OR/MS could bring the two worlds together.

With the present example, we demonstrated that DS can be developed in a relatively simple way for OR/MS applications. Nonetheless, the application of the proposed methodological framework is currently limited to healthcare sector. It is not tested yet in other OR/MS areas such as supply chain, manufacturing, etc. each of which presents different challenges. Also, this implementation involved building the federates anew. A very useful addition would be a detailed activity as to how to modify existing OR/MS simulations in order to become part of a federated DS model. Currently, our OR/MS methodological framework is being used and tested by Saker Solutions (www.sakersolutions.com), a consultancy company that is very active in providing M&S solutions to a big variety of OR/MS applications.

It is recognised that there are conceptual and technical challenges in developing DS. Especially for OR/MS modellers who usually use simulation packages to build simulations, and they rarely are software engineers. OR/MS practitioners and researchers do not develop simulations using low level programming languages and therefore configuring a middleware is not an agreeable task. Especially, while the familiar practices seem adequate even with their limitations, OR/MS modellers would not put effort in adopting DS practices. The main reason is that DS appears technically complicated. In the same time, the benefits of DS in OR/MS applications are not yet clear.

Nonetheless, industrial simulation applications have a lot to gain by employing DS. Some of the benefits are listed below:

- Models for large-scale complex systems can be broken down into smaller subsystem models that interoperate in a federated environment rather than developed as very big OR/MS simulations.
- Models can stay local and be able to run as independent simulations.
- Consistent and up-to-date data; data can be updated for single organisations in one place.
- Data stay private; sharing only the necessary information for interfacing with other federates.
- Models of subsystems are realised using the best fitted simulation technique.
- Connectivity of models is automated and synchronised and supports single-technique and hybrid simulations.
- Cross-platform connectivity, where component models can be developed using different simulation software.
- Reuse of component models to compose different schemes.
- More efficient experimentation by distributing the computational load over a distributed computer infrastructure.

Furthermore, in an OR/MS simulation of a large-scale system that consists of many subsystems, these subsystems are tightly coupled. However, organisations in real world networked operations cooperate in a loosely coupled manner. This can be observed in areas within or outside an organisation where sections are able to share selected information, to communicate and understand each other but in the same time operate independently. A simulation that is able to represent this level of reality and has distributed control over the whole system but supports independency in parts, such as DS, has an advantage and therefore sounds the natural choice for such systems.

Arguably, OR/MS community can benefit from DS. As mentioned earlier, model reusability, repository of composable models that can be used to compose different large-scale distributed simulations can lead to reduced model development time and more efficient use of resources. For example, more experimentation and analysis of results can be conducted due to reduced execution time. Furthermore, with the current rapid development in the field of Big Data and Cloud Computing, composable and interoperable simulations can be linked to other services, such as data models, optimisation components, and visualisation and analytics tools. Lately, efforts have been put on developing cloud platforms for OR/MS simulation packages [49].

Interestingly, recent attempts have been made to simplify the process of developing HLA-based distributed simulations. In the context of the Simulation Exploration Experience (SEE) (www.exploresim.com) project led by NASA, Garro et al. [13] developed a general-purpose, domain-independent framework that aims to ease the development of HLA-based simulations by hiding lots of the technical complexities in developing DS. Their “HLA Development Kit” provides various resources (i.e., software framework, technical documentation, user guide, reference examples, and video-tutorials) for developing HLA federates.

7. Conclusions and further work

In this paper, we presented a methodological framework that can be used as a guide for developing DS from the perspective of the OR/MS practices. We believe that by approaching DS from the OR/MS point of view, DS will become an attractive solution for big simulations. This work will be taken forward by applying the methodological framework to other regional EMS of various sizes and other OR/MS areas such as supply chain and manufacturing. Also, it is planned to demonstrate the practice to simulation experts coming from both DS and OR/MS backgrounds in order to consolidate the approach. We argue that the main limitation at the moment is the lack of simulation theory in the area of DS for OR/MS, which we tried to tackle in this work.

Acknowledgments

This research was funded by the Multidisciplinary Assessment of Technology Centre for Healthcare (MATCH), an Innovative Manufacturing Research Centre (IMRC) funded by the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/F063822/1).

References

[1] L. Aboueljinane, Z Jemai, E. Sahin, Reducing ambulance response time using simulation: the case of Val-De-Marne department emergency medical service, in: Proceedings of the 44th Winter Simulation Conference (WSC12), Berlin, DE, 2012, pp. 943–954, doi:10.1109/WSC.2012.6465018.
[2] A. Anagnostou, A Distributed Simulation Methodology for Large-Scale Hybrid Modelling and Simulation of Emergency Medical Services Unpublished PhD thesis, Brunel University, London, 2014.
[3] A. Anagnostou, A. Nouman, S.J.E. Taylor, Distributed hybrid agent-based discrete event emergency medical services simulation, in: Proceedings of the 45th Winter Simulation Conference (WSC13), Washington, DC, 2013, pp. 1625–1636, doi:10.1109/WSC.2013.6721545.
[4] O.M. Ashour, G.E. Okudan-Kremer, A simulation analysis of the impact of FAHP-MAUT triage algorithm on the emergency department performance measures, Expert Syst. Appl. 40 (1) (2013) 177–187.
[5] J. Banks, J.S. Carson, B.L. Nelson, D.M. Nicol, Discrete-Event System Simulation, third ed, Prentice Hall Inc., USA, 2000.
[6] L. Bernard, A. Wytzisk, U. Streit, Dynamic interoperable geoprocessing and geosimulation – scenarios, frameworks, and prototypes, in: Proceedings of the 4th AGILE Conference on Geographic Information Science, Brno, CZ, 2001.
[7] C.A. Boer, A. de Bruin, A. Verbraeck, Distributed simulation in industry – a survey Part 3 – The HLA standard in industry, in: Proceedings of the 40th Winter Simulation Conference (WSC08), Miami, FL, 2008, pp. 1094–1102, doi:10.1109/WSC.2008.4736178.
[8] S. Brailsford, Modeling human behavior – an (ID)Entity crisis? in: Proceedings of the 46th Winter Simulation Conference (WSC14), Savannah, GA, 2014, pp. 1539–1548, doi:10.1109/WSC.2014.7020006.
[9] A.G. Bruzzone, R. Mosca, R. Revetria, E. Bocca, E. Briano, Agent directed HLA simulation for complex supply chain modeling, Simulation 81 (9) (2005) 647–655.
[10] J. Eatock, M. Clarke, C. Picton, T. Young, Meeting the four-hour deadline in an A&E department, J. Health Organ. Manag. 25 (6) (2011) 606–624.
[11] M. Fakhimi, J. Probert, Operations research within UK healthcare: a review, J. Enterp. Inf. Manag. 26 (1/2) (2013) 21–49.
[12] R.M. Fujimoto, Research challenges in parallel and distributed simulation, ACM Trans. Model. Comput. Simul. 24 (4) (2016).
[13] A. Garro, A. Falcone, N.R. Chaudhry, O.A. Salah, A. Anagnostou, S.J.E. Taylor, A prototype HLA development kit: results from the 2015 simulation exploration experience, in: Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS'15), London, UK, 2015, pp. 45–46, doi:10.1145/2769458.2769489.
[14] A. Hahn, V. Gollücke, C. Buschmann, S. Schweigert, Virtual test bed for maritime safety assessment, Sci. J. Maritime Univ. Szczecin 44 (116) (2015) 116–122.
[15] J.E. Hannay, H.P. Langtangen, C. MacLeod, D. Pfahl, J. Singer, G. Wilson, How do scientists develop and use scientific software? in: Proceedings of the 2nd International Workshop on Software Engineering for Computational Science and Engineering (ICSE 2009), Vancouver, CA, 2009, pp. 1–8, doi:10.1109/SECSE.2009.5069155.
[16] L.B. Holm, F.A. Dahl, Simulating the influence of a 45% increase in patient volume on the emergency department of Akershus university hospital, in: Proceedings of the 42nd Winter Simulation Conference (WSC10), Baltimore, MD, 2010, pp. 2455–2461, doi:10.1109/WSC.2010.5678941.
[17] S. Ibri, M. Nourelfath, H. Drias, A multi-agent approach for integrated emergency vehicle dispatching and covering problem, Eng. Appl. Artif. Intell. 25 (3) (2012) 554–565.
[18] IEEE, Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP), 2010 IEEE Standard 1730-2010.
[19] IEEE, Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP), 2003 IEEE Standard 1516.3-2003.

[20] IEEE 1278.1-2012 - Standard for Distributed Interactive Simulation – Application protocols.
[21] IEEE, Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, 2010 IEEE Standard 1516-2010.
[22] S. Jain, F. Riddick, A. Craens, D. Kibira, Distributed simulation for interoperability testing along the supply chain, in: Proceedings of the 39th Winter Simulation Conference (WSC07), Washington, DC, 2007, pp. 1044–1052, doi:10.1109/WSC.2007.4419703.
[23] S.G. Jones, A.J. Ashby, S.R. Momin, A. Naidoo, Spatial implications associated with using Euclidean distance measurements and geographic centroid imputation in health care research, Health Serv. Res. 45 (1) (2010) 316–327.
[24] K. Katsaliaki, N. Mustafee, S.J.E. Taylor, S. Brailsford, Comparing conventional and distributed approaches to simulation in a complex supply-chain health system, J. Oper. Res. Soc. 60 (1) (2009) 43–51.
[25] U. Klein, T. Schulze, S. Strassburger, Traffic simulation based on the high level architecture, in: Proceedings of the 30th Winter Simulation Conference (WSC98), Washington, DC, 1998, pp. 1095–1103, doi:10.1109/WSC.1998.745965.
[26] C. Laesche, V. Golluecke, A. Hahn, Using an HLA simulation environment for safety concept verification of offshore operations, in: Proceedings of the 27th European Conference on Modelling and Simulation (ECMS 2013), Aalesund, NO, 2013, pp. 156–162, doi:10.7148/2013-0156.
[27] A.M. Law, W.D. Kelton, Simulation Modeling and Analysis, third ed., McGraw-Hill Book Co, Singapore, 2000 ISBN: 0-07-116537-1.
[28] T. Lee, H. Jang, S.H. Cho, J.G. Turner, A simulation-based iterative method for a Trauma center – air ambulance location problem, in: Proceedings of the 44th Winter Simulation Conference (WSC12), Berlin, DE, 2012, pp. 955–966, doi:10.1109/WSC.2012.6465042.
[29] M.E. Lim, A. Worster, R. Goeree, J.E. Tarride, Simulating an emergency department: the importance of modeling the interactions between physicians and delegates in a discrete event simulation, BMC Med. Inform. Decis. Mak. 13 (59) (2013) 1–11.
[30] Q. Long, An agent-based distributed computational experiment framework for virtual supply chain network development, Expert Syst. Appl. 41 (9) (2014) 4094–4112.
[31] C.M. Macal and M.J. North. Tutorial on agent-based modelling and simulation. *J. Simul.*, 4(3):151–162.
[32] A.C. Medina, L.G. Nardin, N.N. Pereira, R.C. Botter, J.S. Sichman, A distributed simulation model of the maritime logistics in an iron ore supply chain management, in: Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH-2013), Reykjavík, IS, 2013, pp. 453–460, doi:10.5220/0004488504530460.
[33] N. Mustafee, M. Sahnoun, A. Smart, P. Godsiff, D. Baudry, A. Louis, Investigating execution strategies for hybrid models developed using multiple M&S methodologies, in: Proceedings of the 2015 Spring Simulation MultiConference (ANSS 2015), Alexandria, VA, 2015.
[34] N. Mustafee, M. Sahnoun, A. Smart, P. Godsiff, An application of distributed simulation for hybrid modeling of offshore wind farms, in: Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS'15), London, UK, 2015, pp. 171–172, doi:10.1145/2769458.2769492.
[35] NASA. NASA Technology Roadmaps: TA 11: Modeling, Simulation, Information Technology, and Processing. Technology Area 11, NASA, Draft, May 2015 online: http://www.nasa.gov/sites/default/files/atoms/files/2015_nasa_technology_roadmaps_ta_11_modeling_simulation.pdf.
[36] B.S.S. Onggo, J. Hill, Data identification and data collection methods in simulation: a case study at the ORH Ltd, J. Simul. 8 (3) (2014) 195–205.
[37] J.A. Paul, L. Lin, Models for improving patient throughput and waiting at hospital emergency departments, Adm. Emerg. Med. 43 (6) (2012) 1119–1126.
[38] G. Pedrielli, M. Sacco, W. Terkaj, T. Tolio, An HLA-based distributed simulation for networked manufacturing systems analysis, J. Simul. 6 (4) (2012) 237–252.
[39] M. Raab, S. Masik, T. Schulze, Support system for distributed HLA simulations in industrial applications, in: Proceedings of the 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation (PADS'11), Nice, FR, 2011, pp. 1–7, doi:10.1109/PADS.2011.5936773.
[40] L. Rabelo, S. Sala-Diakanda, J. Pastrana, M. Marin, S. Bhide, O. Joledo, J. Bardina, Simulation modeling of space missions using the high level architecture, Model. Simul. Eng. 2013 (2013) 12 967483.
[41] M.H. Rahmat, M. Annamalai, S.A. Halim, R. Ahmad, Agent-based modelling and simulation of emergency department re-triage, in: Proceedings of the IEEE Business Engineering and Industrial Applications Colloquium (BEIAC), Langkawi, MY, 2013, pp. 219–224, doi:10.1109/BEIAC.2013.6560119.
[42] S Robinson, Simulation – The Practice of Model Development and Use, John Wiley & Sons Ltd, Chichester, UK, 2004.
[43] R. Siegfried, J. Luethi, G. Herrmann, M. Hahn, A comparison of DSEEP with the German approach VEVA, in: Proceedings of the 2011 Spring Simulation Interoperability Workshop (SISO), Boston, US, 2011.
[44] SISO, Standard for Commercial-off-the-shelf Simulation Package interoperability reference Models, 2010 SISO-STD-006-2010.
[45] R. Steinbach, J. Cairns, C. Grundy, P. Edwards, Cost benefit analysis of 20 mph zones in London, Inj. Prev. 19 (3) (2012) 211–213.
[46] S. Strassburger, T. Schulze, R. Fujimoto, Future trends in distributed simulation and distributed virtual environments: results of a peer study, in: Proceedings of the 40th Winter Simulation Conference (WSC08), Miami, FL, 2008, pp. 777–785, doi:10.1109/WSC.2008.4736140.
[47] S. Strassburger, T. Schulze, M. Lemessi, Applying CSPI reference models for factory planning, in: Proceedings of the 39th Winter Simulation Conference (WSC07), Washington, DC, 2007, pp. 603–609, doi:10.1109/WSC.2007.4419653.
[48] H. Taghaddos, S.M. AbouRizk, Y. Mohamed, I. Ourdev, Distributed agent-based simulation of construction projects with HLA, in: Proceedings of the 40th Winter Simulation Conference (WSC08), Miami, FL, 2008, pp. 2413–2420, doi:10.1109/WSC.2008.4736349.
[49] S.J.E. Taylor, A. Anagnostou, T. Kiss, G. Terstyanszky, P. Kacsuk, N. Fantini, A tutorial on cloud computing for agent-based modeling & simulation with REPAST, in: Proceedings of the 46th Winter Simulation Conference (WSC'14), Savannah, GA, 2014, pp. 192–206, doi:10.1109/WSC.2014.7019888.
[50] S.J.E. Taylor, S.J. Turner, S. Strassburger, N. Mustafee, Bridging the gap: a standards-based approach to OR/MS distributed simulation, ACM Trans. Model. Comput. Simul. 22 (4) (2012) 18.
[51] Z. Tu, G. Zacharewicz, D. Chen, A federated approach to develop enterprise interoperability, J. Intell. Manuf. 27 (1) (2016) 11–31.
[52] Z. Tu, G. Zacharewicz, D. Chen, Building a High-level architecture federated interoperable framework from legacy information systems, Int. J. Computer Integr. Manuf. 27 (4) (2014) 313–332.
[53] O.M. Ulgen, J.J. Black, B. Johnsonbaugh, R. Klungle, Simulation methodology–a practitioner's perspective, Int. J. Ind. Eng. Appl. Pract. 1 (2) (1994). online (last accessed May 2016)http://www.pmcorp.com/Portals/5/_Downloads/SimulationMethodology_APractitioner'sPerspective.pdf.
[54] M. van Buuren, K. Aardal, R. van der Mei, H. Post, Evaluating dynamic dispatch strategies for emergency medical services: TIFAR simulation tool, in: Proceedings of the 44th Winter Simulation Conference, Berlin, DE, 2012, pp. 509–519, doi:10.1109/WSC.2012.6465214.
[55] Y. Wang, K.L. Luangkesorn, L. Shuman, Modeling emergency medical response to a mass casualty incident using agent based simulation, Socio-Econ. Plann. Sci. 46 (4) (2012) 281–290.
[56] L. Wang, An agent-based simulation for workflow in emergency department, in: Proceedings of the 2009 IEEE Systems and Information Engineering Design Symposium (SIEDS'09), Charlottesville, VA, 2009, pp. 19–23, doi:10.1109/SIEDS.2009.5166148.