International Journal of Production Research, 2016
Vol. 54, No. 23, 7212–7230, http://dx.doi.org/10.1080/00207543.2016.1221535

A service-oriented simulation integration platform for hierarchical manufacturing planning and control

Dong Xu, Sai Srinivas Nageshwaraniyer and Young-Jun Son*

Systems and Industrial Engineering, The University of Arizona, Tucson, AZ, USA

(*Received 30 October 2015; accepted 29 July 2016*)

In this paper, a coherent and comprehensive distributed simulation platform is proposed to support hierarchical manufacturing planning and control. This platform enables integration of various hardware and software components within and across supply chain members, such as manufacturing equipment, physics-based process simulators, system-dynamic, agent-based and discrete-event simulators (DESs), and databases via web services technology. At the shop level, a DES model may run in a stand-alone mode, or in conjunction with other simulators (e.g. process or agent-based simulators) and/or hardware (e.g. robots and machines). At the enterprise level, multiple shop floor simulators (each representing a factory) are integrated into a geographically dispersed environment. Exemplary models simulating aircraft drill and assembly on the shop floor are developed in widely used packages, such as Arena®, Simio®, Anylogic® and ESPRIT®, and then integrated. The proposed platform with the hardware-in-the-loop capability is successfully implemented and demonstrated for an automated manufacturing cell in the Computer Integrated Manufacturing and Simulation lab at the University of Arizona. Experiments are conducted using the proposed platform to test (1) reduction in estimated variance of part cycle time given variations in order arrival rate, effectiveness of materials processers and material handlers, and (2) computational time performance of web services. The experimental results reveal that the proposed platform is viable to enable both vertical (shop simulator to equipment level process simulators and/or equipment) as well as horizontal (e.g. multiple shop level simulators) integrations.

Keywords: distributed simulation; process simulation; simulation integration; manufacturing planning and control

1. Introduction

With advances in the means of transportation and communication, the concept of supply chain has transcended into the global supply chain. This is especially true for the products requiring specialised manufacturing processes carried out in dedicated factories located in different areas of the world that may be owned by different companies. Industries such as aircraft, automotive and computers are prime examples, and are composed of suppliers, manufacturers and customers spread all across the globe. Meeting their time and cost objectives by planning, tracking and controlling various product life cycle activities can be made easier through technology-enabled solutions.

Due to the involvement of many complex and dynamic subsystems, simulation is often considered as an appropriate tool for the modelling, evaluation and planning for a large supply chain (van der Zee and van der Vorst 2005). Since it is reasonable to analyse a large system by considering it as a system-of-systems, the global supply chain can be decomposed into various subsystems, which interact with each other as and when needed. Distributed simulation technology (Misra, Venkateswaran, and Son 2003) is promising for modelling such type of subsystems and their interactions. A major need for distributed simulation applications on global supply chain simulation is motivated by the intellectual property protection policy (Venkateswaran and Son 2004; Strassburger, Schulze, and Fujimoto 2008). A monolithic supply chain simulation model is very often difficult to be built, since each company in the supply chain is averse to sharing its proprietary information containing processes' details. Distributed simulation technology can facilitate the integration by connecting each member's model as a black box into the overall distributed supply chain simulation (Strassburger, Schulze, and Fujimoto 2008). Under such a scenario, the coordination of the activities of different members is supported via pre-agreed and member-based (making information available to particular member only) release of the information and process details. Even in cases where confidentiality is not an issue, rapid integration of currently existing and dispersed simulation models is an advantage offered by distributed simulation technology as opposed to

*Corresponding author. Email: son@sie.arizona.edu

combining all models into a monolithic model whose development might require large amounts of time and cost. Another benefit of employing the distributed simulation technology is reduction in overall simulation execution time, sharing of workload of entire simulation and enhanced tolerance of simulation failures (Terzi and Cavalieri 2004). In many cases, enhanced problem identification, exploration, as well as further analysis of the manufacturing process are facilitated by physics-based process simulators or hardware-in-the-loop simulators. In addition, if hardware-in-the-loop is established, control can also be performed.

The major achievements of service-oriented architecture (SOA) involve reuse of IT technology or systems leading to cost reduction, and thereby generating return-on-investment (Adams and McNamara 2006). One of important issues nowadays in the manufacturing control system on the shop floor is the reuse of legacy frameworks developed years ago. System re-development with the same functionality as that of the legacy framework is not a viable option since it might have to be repeated over time. SOA provides solutions to integrate legacy frameworks for reusing them without much re-development efforts involved. About 67% qualified respondents reported that SOA helps the integration of systems with different ages; other reported benefits include more flexible architecture (71%), data integration (62%) and service integration (59%). Also, service-oriented development of applications can reduce the total IT expenses as much as 20% over the long run, when comparing with the traditional client–server development approaches, with the savings amount growing exponentially over time as the service library expands.

In a global supply chain, the shop level operations not only have to achieve time and cost objectives, but also have to be coordinated with other factories for maintaining the overall supply chain productivity. Failures in such coordination result in delay in the delivery of the final product, and consequently, heavy penalties/loss of contracts. A famous example of such delayed deliveries is Boeing's Dreamliner 787 aircraft, which was launched three years after the originally scheduled delivery date. Several postponements that eventually lead to delay were attributed to various reasons (Ray 2008; Ostrower 2011; Ferrari 2011) such as (1) supplier part shortages, (2) unfinished work by suppliers, (3) engineering/manufacturing changes and (4) fastener mismatch and redesign. The Boeing 787 Dreamliner team struggled to update 787's overall programme schedule after labour strike and part shortage issues occurred in one factory (Ray 2008). Another setback occurred when electrical fire and power failure originated at the engine during an aircraft testing process while in-flight (Ferrari 2011). In summary, we note that avoiding such delays in the future would require the following actions: (1) improving data estimation by incorporating prediction capability via digital simulation of those manufacturing processes especially involving hundreds of thousands of small components like fasteners (Gunsalus 2007), and (2) using estimated data to devise alternative process plans which upon implementation can improve supply chain coordination by reducing the impact of the unforeseen delays. In this paper, the aircraft supply chain is chosen as case study as not only it is global, but as it would benefit from planning for coordination-related problems among its members in facing delays.

In this paper, a coherent and comprehensive service-oriented simulation integration platform is proposed, which integrates heterogeneous modules (hardware and software) in a global supply chain via web services to support simulation-based real-time planning and control. At the shop level, both physics-based process simulators as well as agent-based simulators are integrated together with the discrete event simulator (DES) for planning; hardware is integrated with DES for real-time planning and control. High-fidelity simulation is made possible by employing process simulators that simulate machining processes as well as employing agent-based simulator for simulating workers; hardware can also be employed for increasing fidelity further. At the enterprise level, multiple shop floor simulators belonging to different factories along with a system dynamics simulator are integrated and run in a time synchronised manner. Web services technology serves as a transaction coordinator for the interactions among heterogeneous simulations in the federation of multiple DESs, agent-based, system dynamics and process simulators, along with any hardware-in-the-loop. Web services technology is employed to facilitate two directions of integration: (1) horizontal integration (e.g. shop simulator to shop simulator) and (2) vertical integration (e.g. shop simulator to hardware, shop simulator to process simulator, shop simulator to agent-based simulator, or shop simulator to system dynamic simulator). The proposed platform is coherent in terms of utilising the web services for time synchronisations and data interactions. These horizontal and vertical integrations facilitate the information sharing and operations' coordination among the enterprise level, the shop level and the equipment level.

Much of the contents in this paper have been updated from a dissertation work (Xu 2014), which is available online. Readers are suggested to find more details about the relevant topics in the dissertation. This paper combines and extends the extensive work available in the literature – many performed by a co-author of this paper – on simulation-based planning and control into a comprehensive and coherent platform through which integration between multi-paradigm simulations, high- and low-fidelity simulations, legacy applications along with hardware-in-the-loop is made possible. First, the proposed service-oriented simulation integration platform has been inspired by the one proposed in Xu and Son (2014) for production planning and control. Its extension for a shop floor planning and control system (Smith et al. 1994; Wysk, Peters, and Smith 1995; Son et al. 2002) is discussed in detail in this paper. Second, the details of the simulation models at the shop and enterprise levels have been formalised extending information models presented in Son, Jones, and Wysk (2003). An exemplary resource model has also been provided for the aircraft wing assembly under the considered case study (Xu et al. 2011). Third, main controller functions at the enterprise, shop and equipment levels executed through web services have been briefly discussed; for this purpose, these functions available in Cho, Son, and Jones (2006) are adopted. Fourth, the sequence of interactions between shop and equipment level components along with the enterprise level order execution details (Venkateswaran and Son 2005; Lee, Son, and Wysk 2007) have been formalised. Finally, the platform has been implemented to demonstrate planning as well as control at the shop level as well as enterprise level. Figure 1 depicts various system configurations that the proposed platform is intended to support. In Figure 1, X axis labels are hardware and software, Y axis labels are control and planning and Z axis labels are shop and enterprise levels. For example, the origin in Figure 1 refers to the case, where software (e.g. real-time simulator) is used to drive/control a manufacturing cell. Through extensive experiments (corresponding to the non-origin data point in Figure 1), planning for reduction of estimated variance in part cycle time under variations in the order arrival rate, effectiveness of material processors and material handlers at the enterprise level is analysed. The platform is also shown to support large simulation federation sizes involving many interactions and frequent time synchronisations in the federation.

Although there have been many works on individual topics in the literature that have also been briefly discussed in this paper such as simulation-based planning and control, distributed simulation involving high- and low-fidelity simulations, integration of legacy software into a shop floor planning and control platform, there has not been any work that proposes to realise the advantages of combining these methods into a coherent and comprehensive platform for planning as well as control in a manufacturing enterprise (see Figure 1). The proposed platform has the following contributions: (1) enabling seamless transition not only between planning and control (i.e. across functions) but also between enterprise and shop levels (i.e. across levels) by being able to switch between use of software comprising simulators of differing fidelity and paradigms, and hardware; and (2) enabling detailed status estimation via physics-based process simulators and hardware in order to enhance the accuracies of the simulation outputs that are later used for planning as well as control.

The remainder of this paper is organised as follows: In Section 2, the literature work related to the proposed platform is summarised. In Section 3, the details of constituents of the proposed hierarchical distributed simulation integration platform are discussed. In Section 4, the system implementations at the enterprise, shop and equipment levels are discussed. In Section 5, experimental scenarios are constructed and results are then presented. Finally, in Section 6, the conclusions and future extensions to the platform are discussed.

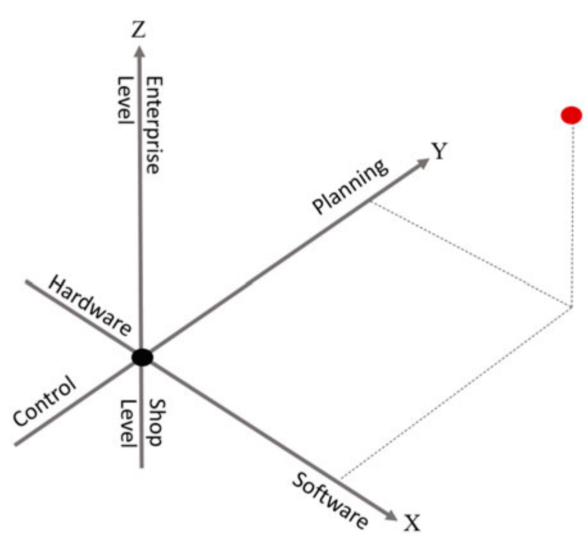

Figure 1. Various system configurations supported by the proposed platform.

2. Background and literature survey

Most of the current literature reviews are from a dissertation work (Xu 2014), which is available online. Readers are suggested to find more details about the relevant topics in the dissertation. Researchers have widely studied the structures, benefits, challenges and other related issues of SOA. For example, Schmidt et al. (2005) presented the essential features such as key components, integration model and user roles of Enterprise Service Bus, which manages and supports meta-data through usage patterns. Various usage patterns were described in details such as gateway pattern, event distribution pattern, service transformation pattern and matchmaking pattern. Bieberstein et al. (2005) explored various challenges such as governance, economic and enterprise to SOA-based IT transformation. The paper emphasised IT systems, organisational structures, cultural transformation, and associated behavioural practices. Later, Papazoglou and van den Heuvel (2007) discussed Enterprise Service Bus with various functions and associated components. The paper also extended conventional SOA for additional capabilities, including service orchestration, provisioning, message security and service management. Bozkurt, Harman, and Hassoun (2009) conducted a survey related to the testing and verification of SOA, where various testing perspectives were considered such as service-centric system, unit, fault, model, interoperability, integration, collaborative, QoS and regression-based tests. Later, Guo (2012) studied standard approaches for simulation service composition in the service-oriented computing environment. A layered-framework was proposed including infrastructure, execution engine, simulation service, simulation experiment and graphical user interface layers; a wildfire-spread simulation was used as an application in that work.

SOA has been studied by several researchers for the manufacturing/production planning and control as well. Karnouskos et al. (2007) proposed web services-based enterprise system integration via SOA-ready networked embedded devices. The paper also discussed the benefits such as reduction in production costs and enabled manufacturing flexibility. Similarly, Alvares et al. (2008) discussed a web-based manufacturing management and control system, where a customer can utilise manufacturing services via SOA. Furthermore, a methodology was proposed to integrate different parts of product engineering via ERP; the implementation of a web-based shop floor controller was also discussed. Later, Jacobi, Hahn, and Raber (2010) presented an integrated SOA and multi-agent system approach via a model-driven method, where the transformation from SoaML to DSML4MAS was utilised. The paper further demonstrated the effectiveness of the proposed approach in terms of increased flexibility during planning and scheduling for a steel manufacturer. Ollinger, Schlick, and Hodek (2011) focused on the methods and potential for transferring SOA from IT to automation. A process-oriented production planning method was illustrated, and a technical demonstrator with a purpose to evaluate SOA technology was also discussed in the paper. Mendes et al. (2012) discussed the benefits of SOA in manufacturing system to achieve modularity, flexibility, re-configurability and interoperability by leveraging Petri net formalism. Zhang et al. (2012) proposed service-oriented, cross-platform and high-level machining simulation platform to provide interoperability of multiple numerical control (NC) machining environments, which makes the operation training easier and enables the networked manufacturing development. Later, Valilai and Houshmand (2013) had developed a service-oriented collaborative manufacturing platform named XMLAYMOD, supporting XML data structures. This platform was able to fulfil the requirements of distributed collaboration and data integration based on the STEP standard. Zhong et al. (2013) proposed an RFID-enabled real-time advanced production planning and scheduling shell, which utilised a Software as a Service model as well as SOA. The shell also leveraged extensible markup language-based model for the easy-to-deploy capability as well as other solution algorithms and scheduling rules that were deployed as web services. Later, Tao et al. (2014) studied a cloud computing and Internet of Things-based cloud manufacturing service system, where the architecture and technology for realising such a system were also discussed. However, a comprehensive and coherent system is needed to support various decisions across the service-oriented platform.

Over the recent two decades, distributed simulation has been applied to support analysis and decision-making in supply chains. Its applications have appeared in military as well as in industries such as semi-conductor industry (Lendermann et al. 2003) and automotive industry (Taylor et al. 2002). In general, it is suitable for military or industry areas having the following characteristics (Lendermann et al. 2003): (1) high variability and stochastic conditions in the production/work environment; (2) complex task dependency, and operations including large amount of handshaking processes; and (3) possibility of structural changes in the supply chain. It's a very useful tool to set up realistic settings of a supply chain.

One of the first distributed simulations was proposed in Chandy and Misra (1979), where simulators communicate via messages with neighbours, and deadlock was avoided by proving the correctness of each of the system component processes. Later, the concept of distributed simulation in the context of discrete-event simulation was proposed in Misra (1986). A somewhat recent summary and future trends of architectures and standards in distributed simulation can be found in Strassburger, Schulze, and Fujimoto (2008). Among all the parallel and distributed simulation architecture, High Level Architecture (HLA), originally developed by the US Department of Defence, is currently the most popular one, and is defined under IEEE standard 1516. In HLA, the interactions between simulations are achieved via Run-Time

Infrastructure (RTI). A federation is defined as the integration of multiple simulation models connected via RTI, and each simulation model in it is referred to as a federate. A standard Object Model Template is used by HLA as described in Lutz, Scrudder, and Graffagnini (1998). The times of various federates are synchronised through services provided by HLA (Fujimoto 1998). Individual simulation model details are hidden, and only the required information is exchanged via interactions (Kuhl, Weatherly, and Dahmann 1999). Other common terminologies involved in HLA include object, attribute, interaction and parameter, interface specification, and HLA rules and their definitions can be found in Dahmann, Kuhl, and Weatherly (1998). Several drawbacks of HLA summarised in Strassburger, Schulze, and Fujimoto (2008) are as follows: no load-balancing mechanism, poor scalability, supports only syntactic interoperability, no semantic interoperability, complex standard structure and time consuming to adopt and maintain.

To overcome the drawbacks in distributed simulation, especially the lack of interoperability, a state-of-the-art technology called web services technology was used in Rathore et al. (2005) and Lee, Son, and Wysk (2007), to create a message-centred distributed simulation. According to the World Wide Web Consortium (W3C), web service is defined as 'a software system designed to support interoperable machine-to-machine interaction over a network'. In this paper, the machine is a computer; and communication takes place between digital simulation models running in computers connected via the Internet. The web services follows the prototype of standard protocol stack including HTTP serving as data transport media, SOAP XML for message format, WSDL for web services function description and UDDI for publishing the service online. Web services technology was demonstrated in Rathore et al. (2005) and Lee, Son, and Wysk (2007) to possess both the flexibility for customising time synchronisation methods, and the simplicity required for use. In this paper, it is aimed to enhance and re-use the web services technology so that it can be used to integrate not only DESs but also process simulators such as ESPRIT® as well as hardware in the shop floor.

Two major functionalities offered by federation services are (Riley et al. 2004): (1) time synchronisation; and (2) interactions. The existing methods for time synchronisation can be classified into three categories: (1) conservative; (2) optimistic; and (3) hybrid. A summary of these approaches can be found in Fujimoto (2001). A technical discussion to unify the different time advance approaches provided by HLA-RTI can be found in Huang et al. (2005). In this paper, an epoch time synchronisation approach, which first appeared in Rathore et al. (2005), is adopted and implemented. Several drawbacks of both conservative and optimistic time synchronisation approaches can be overcome by this approach. A look-ahead function that is embedded into each federate is used to estimate the nearest epoch event time in the distributed simulation. It was proven that the by using this approach, primarily, simulation runtime and the number of requested time synchronisation steps were significantly reduced. As shown in the literature, different factors such as type of system being modelled and available computational resources can impact the performance of distributed simulation and transaction coordinator used for time-synchronisation of different simulators.

3. Proposed simulation-based integration platform of manufacturing planning and control

The service-oriented platform in this work has been developed according to the following steps: service identification, specification, composition, realisation and implementation (Arsanjani 2004). The service identification is responsible for establishing a service structure and capabilities. The service specification is committed to identifying the service interfaces and the service data model. The service composition discusses the service choreography of different participants, and the service realisation and implementation concern the actual development of a service-oriented system. The detailed discussion of each of these steps in the overall development of the platform can be found in Xu and Son (2014). Figure 2 shows a sample service decomposition diagram in a manufacturing process, from the enterprise level to the shop floor level, and then further extended to the cell (equipment) level. Each level includes a group of services, which in turn can be further decomposed into sub-services. The services are denoted by circles or oval shapes boxes, and their operations are denoted by rectangular-shaped boxes. The proposed platform has the capability to integrate applications running into current frameworks with legacy frameworks as well.

3.1 *Shop level planning and control*

A simulation model of an enterprise is composed of multiple shop level simulation processes (e.g. supply chain participants such as supplier, assembly plant and transporter). Figure 3 shows the proposed service-oriented simulation integration platform for a shop floor planning and control system. Here, a DES performs the role of planner/scheduler or real-time controller, as needed. When the DES performs planning/scheduling, process simulator is used at the Equipment level instead of a real piece of equipment (e.g. robot, milling machine or turning machine). Similarly, agent-based models could be used for simulating human workers. Communications between simulators in different levels are supported via web service. A database is implemented in MS SQL server; and the controller is a software. When the DES

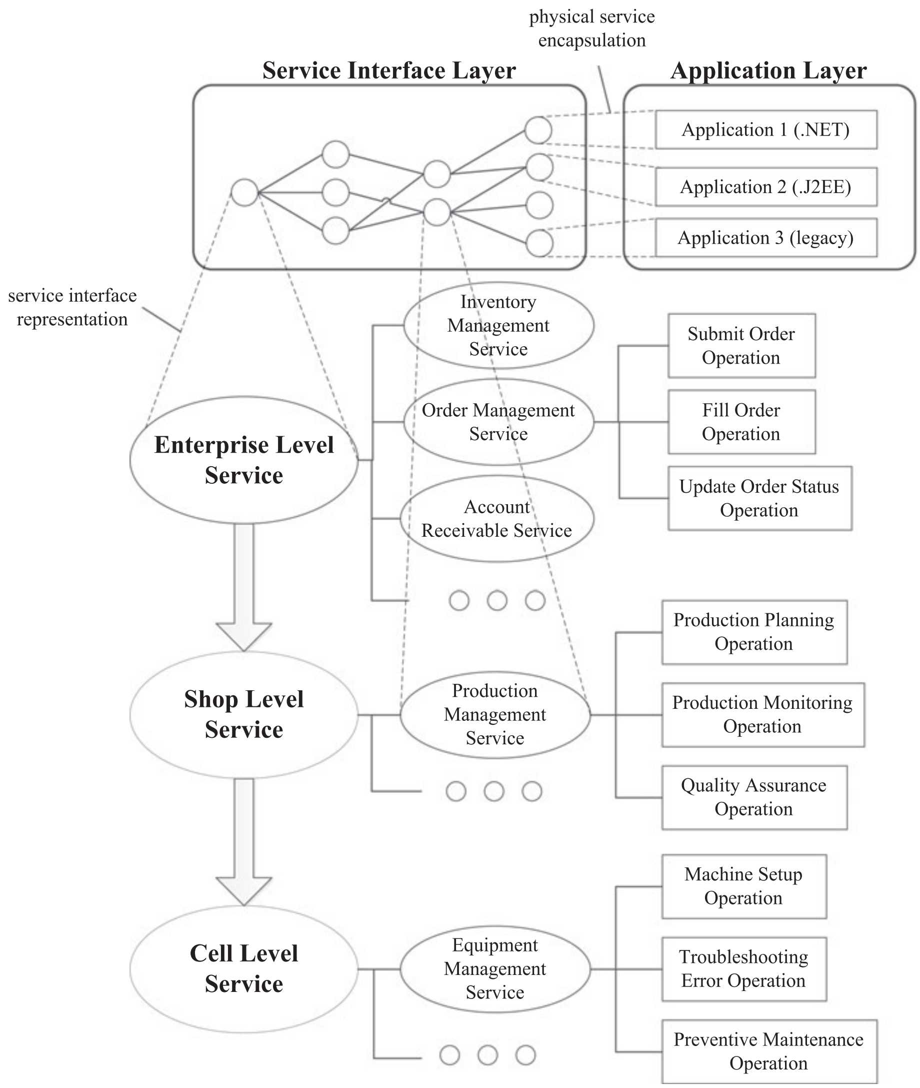

Figure 2. Service decomposition diagram (Xu and Son 2014).

is used as a real-time controller, it sends tasks to be accomplished via web services to the controller; and in turn it receives their status updates.

Figure 4 depicts an exemplary service choreography diagram for a shop floor control system involving DES, web service, SQL server, process simulator, controller software and a unit control device (UCD). In this time-sequence diagram, the interactions of DES – which acts as the shop supervisor – are as follows: (1) retrieve process data from SQL server; (2) pass on the process data to the process simulator via web services for evaluation until an optimal plan is obtained based on objectives such as minimum processing time while minimising tool degradation; (3) communicate this plan via web services through messages sent to the controller software that in turn sends machine commands to the appropriate machine on the shop floor to execute all the tasks involved. This figure could be redrawn for a choreography between a supervisor (DES) and human, where the human is simulated by an agent-based model. The discussion regarding appropriate replacements for the controller software and UCD objects in the sequence diagram for the service choreography between DES and human is out of scope for this paper.

3.2 *Enterprise level planning and control*

The enterprise level order execution in this work is represented via finite state automata graph as shown in Figure 5. This graph illustrates a sample procedure of receiving, loading, processing and shipping an order. Figure 6 depicts how

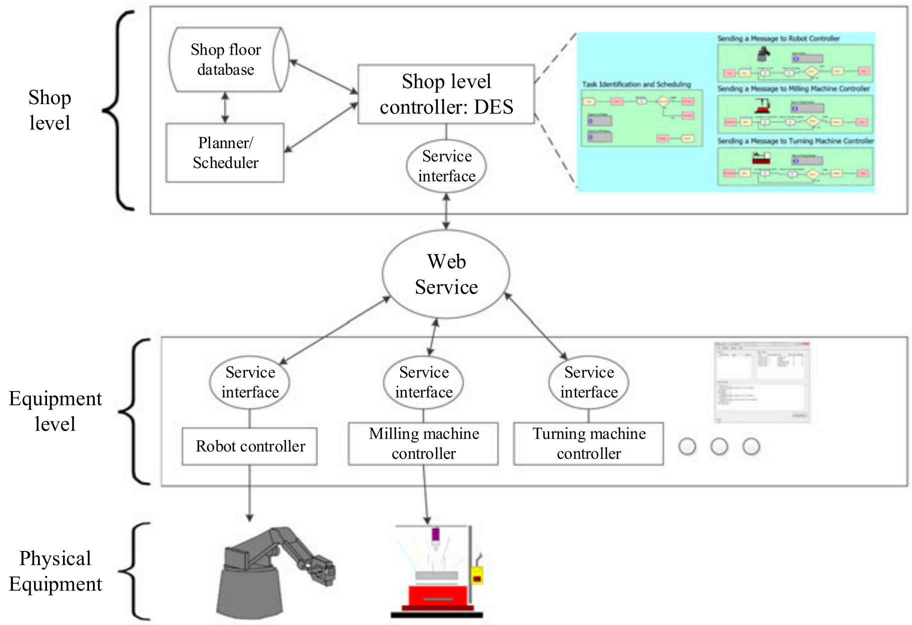

Figure 3. Illustration of service-oriented simulation integration platform for a shop floor control system.

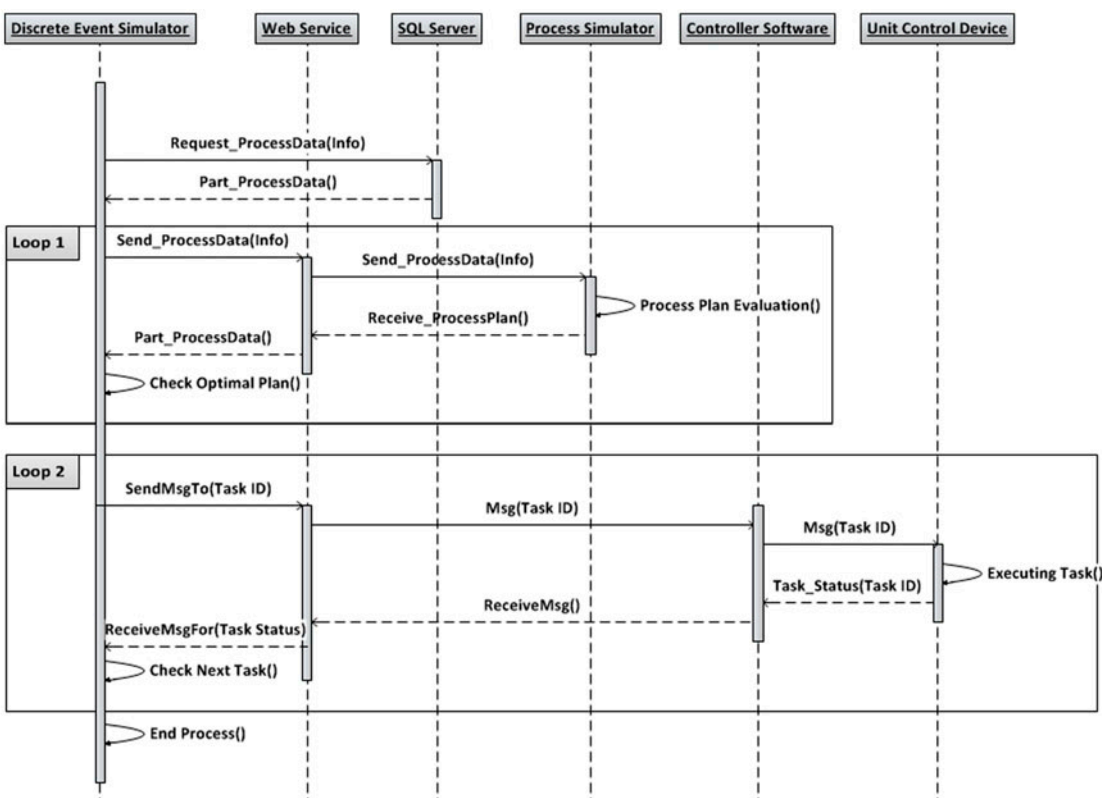

Figure 4. Exemplary service choreography diagram for a shop floor.

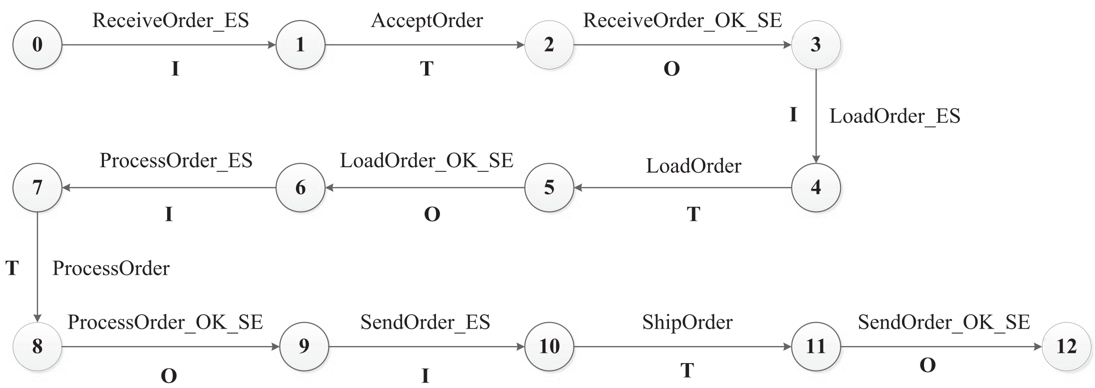

Figure 5. Finite state automata graph for an enterprise level execution model (Venkateswaran and Son 2004; Lee, Son, and Wysk 2007).

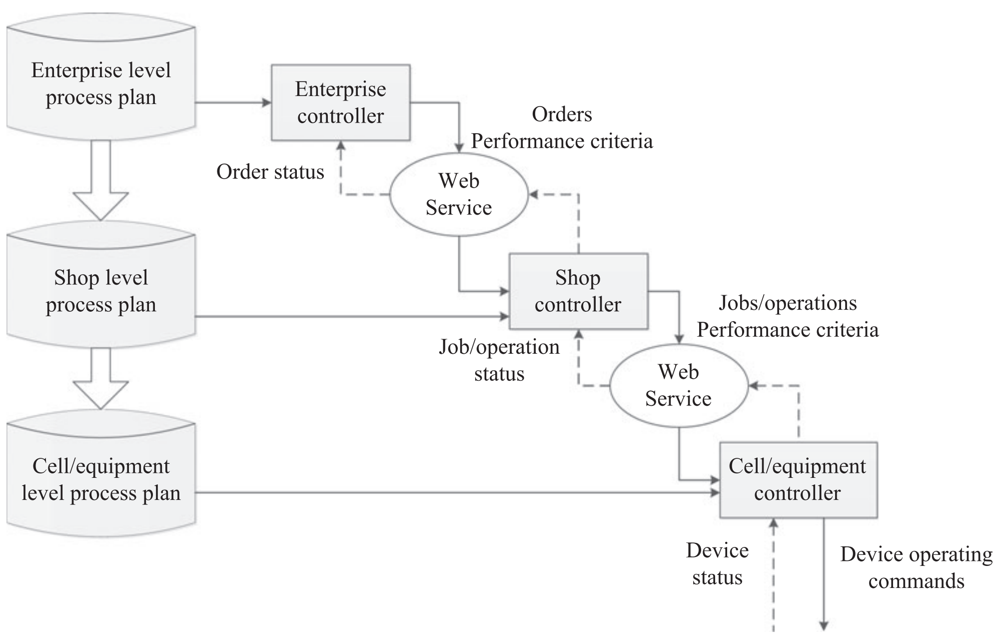

Figure 6. System interactions using web services among different control levels (Cho, Son, and Jones 2006).

controllers at each level interact with an upper or lower-level controller via web services. Different information are transferred via web service, for example, the order status information is fed to the enterprise level controller by the shop level controller, and the enterprise level controller can provide the order and performance criteria information to the shop level controller. From the top-down view, the enterprise level process plan is further decomposed into the shop level process plan, and the shop level process plan is further decomposed into equipment level process plan. Data flow containing either the process plan or orders/performance criteria/commands are illustrated by normal arrows. The direction of these arrows is either from the database to controller or from upper-level controller to lower-level controller. The data flow representing order/job or device status updates is represented by dashed arrows. The direction of these arrows is always from lower-level controller to upper-level controller.

4. System implementation

Most of the system implementation in this section are from a dissertation work (Xu 2014), which is available online. Readers are suggested to read more details about the relevant topics in the dissertation.

4.1 *Equipment level simulator*

A process simulator is used to simulate machine processes. In this paper, ESPRIT® has been employed to estimate the machine processing times for a process plan (i.e. a series of NC code) given a set of machining parameters. They include parameters such as machine type, work-piece material, cutting tool, feed rate, cutting velocity and depth of cut. The machining parameters are stored in tables within Microsoft SQL Server 2008®, and then retrieved by the process simulator for the estimation purposes. The process simulator can be used to evaluate alternative process plans as well. During the evaluation, the process simulator provides flexibility with respect to the tool and machine selections for each machining process. Figure 7 shows the schematic diagram of the shop floor data collection and process plan evaluation by ESPRIT®.

In general, there are two issues regarding the data and process management in the process simulation that have to be resolved before or during the course of simulation: unavailability of data and unsuitability of recommended parameters by the process simulator. First, since not all required data are readily available, some of the data is processed in a way that the unavailable data can be estimated from the available data. One technique is extrapolation using mathematical equations with the available data as inputs and estimation for the unavailable data as outputs. For example, the variation in the surface finish of a work-piece – due to tool degradation – is assumed to be a function of tool's age and the expected tool life. Second, the current machining parameters such as feed rate and cutting speed that could be updated without disrupting the shop floor, overridden by the values recommended by the online repository knowledge-base of ESPRIT®. This is especially useful when the parameters recommended by the simulator go beyond the capability of the machine and tools available on the shop floor. In such a case, a plan with parameters best suited to the resources on the shop floor can be retrieved from the online repository.

4.2 *Shop level simulator*

Figure 8 shows the shop level simulation information required to build a simulation model, which contains different levels of information such as product process information, production information, station schedule information, output information, header information and shop floor information. The product process information contains the product-related process plan information and product information, whereas the process plan information contains process flow information and operation information. Furthermore, the process flow information includes immediate future information

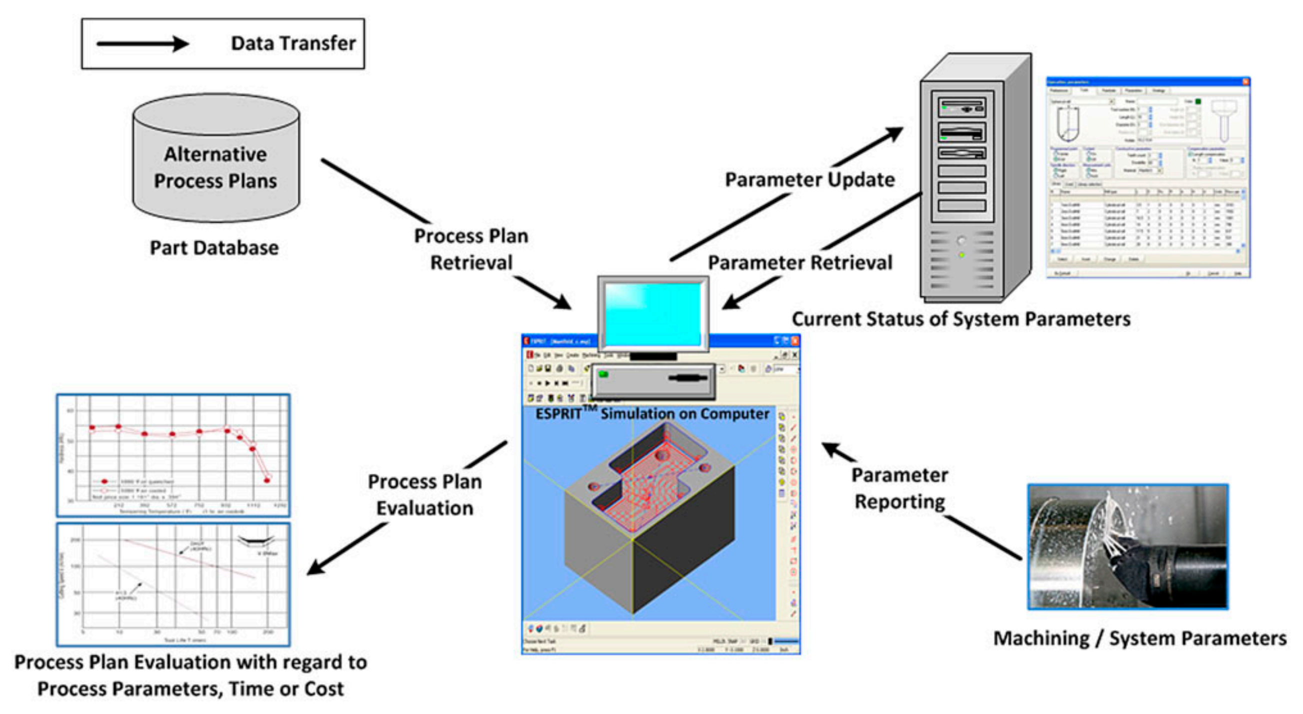

Figure 7. Schematic diagrams for a process simulation (Xu 2014).

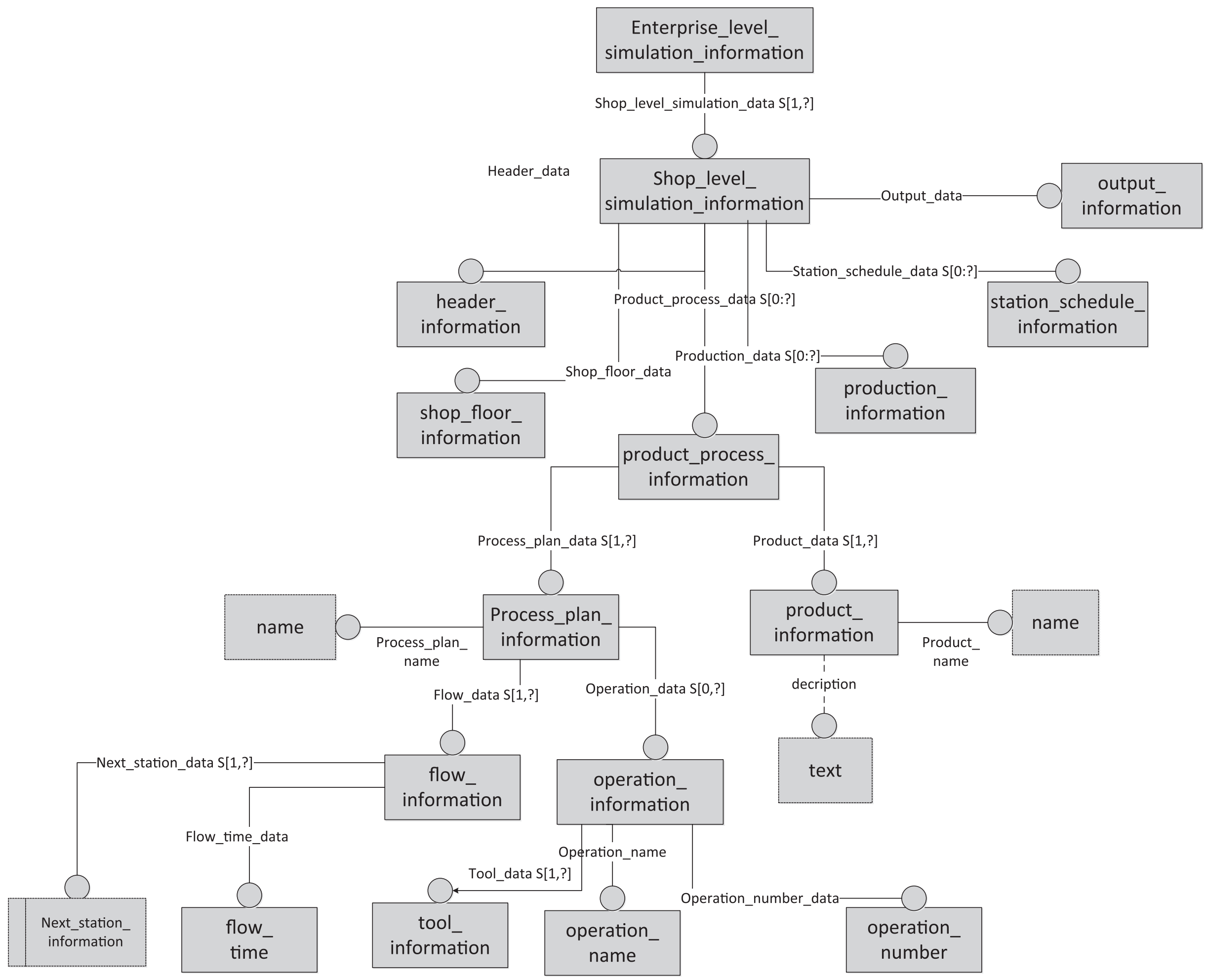

Figure 8. Information model for shop level simulation (Son, Jones, and Wysk 2003; Xu 2014).

such as next station and estimated flow time; the operation information includes tool, operation name and index information. Son, Jones, and Wysk (2003) have presented a similar information model schema.

At the shop floor, the resource model involves a hierarchical structure starting from the shop floor level, all the way down to the detailed level including machine, tool properties and requirements. Steele, Son, and Wysk (2001) discussed resource modelling for the integration of the manufacturing enterprise. An exemplary resource model for an aircraft wing drilling shop floor used in this work is shown in Figure 9, where four types of equipment (material processor, material handler, material transporter and buffer storage) are provided. In the wing drilling shop floor, wing hole drilling process is performed by direct drilling tools, which are held by different types of flex track. Three different types of flex track are used in this shop: Flex Track, HD Flex Track and Mini Flex Track; they all inherit the properties from the holding tool class. The flex track tool specifications such as tool type, weight, dimension, feed, depth of cut and moving speed are also provided. Under the material handler class, a robot is mainly responsible for setting up different flex tracks, while human is responsible for machine control and error inspection. The material transporters such as conveyor and AGV are responsible for transporting the raw materials into the shop, as well as transporting the finished parts into the next production station. Xu et al. (2011) utilised a design structure matrix and quantitative risk measurement based on a similar resource model for estimating aircraft design change propagation effects.

The shop level simulator interacts with machines and robots via web services. Various machine movements are governed by an Arena® simulation model at the supervisory level with the equipment level controller. At the equipment level, the controller communicates with the UCD via bi-directional serial communications through COM ports. For each task, the controller sends the name of the related NC file, robot advanced control language programme or laser-cut file

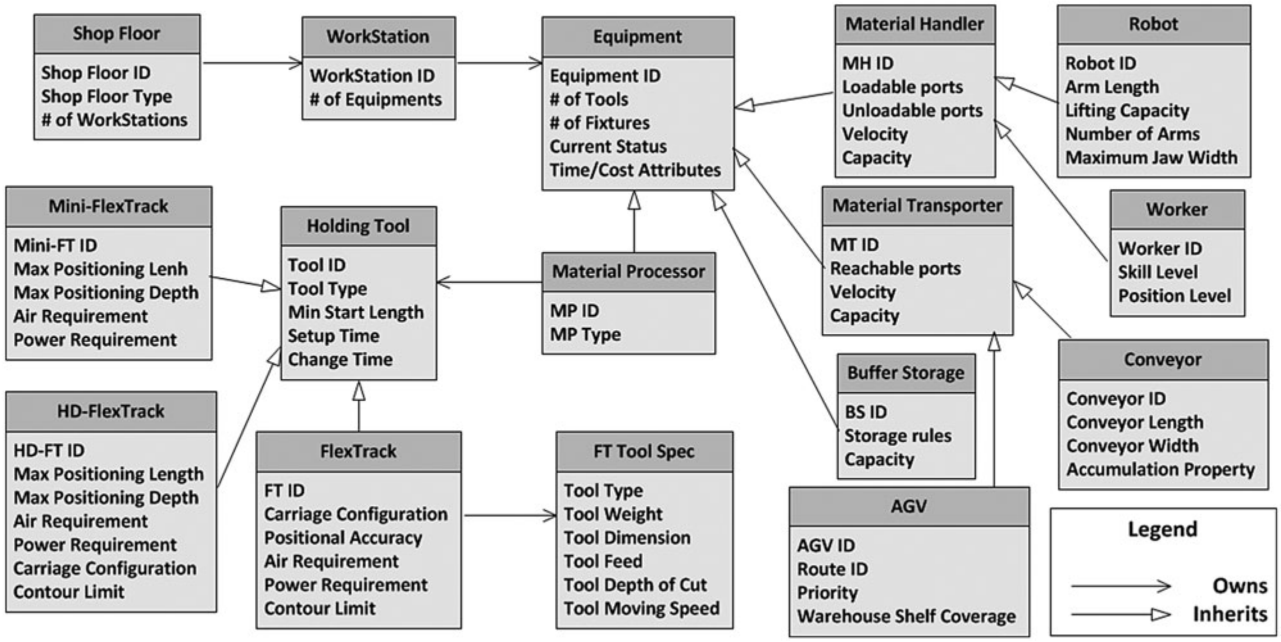

Figure 9. An exemplary resource model for an aircraft wing drilling shop (Xu 2014).

to the equipment UCD for execution. After task completion, UCD sends a completion message to the equipment controller, which in turn reports the task status to the Arena® supervisor via web service.

There are two main functions that allows Arena® simulation model to serve as a supervisor. The first function is 'Initialization', in which the supervisor reads product ID, product quantity and process plan from the SQL Server database. The other function is 'Task Generation', which involves sending task-related messages to the machine/robot controllers via web services. Figure 10 describes the process flow diagram of simulation-based shop floor control containing the Arena® supervisor (a) and an equipment level controller (b). After a message is sent, the supervisor repeatedly checks the message queue in the web service looking for a possible response from the equipment controller. In the Arena® model, the 'Delay' block is used to create the time interval required between successive message checking events. Once a task completion message is received by Arena®, the message for next task – if available–will be sent to the equipment controller. The supervisor goes through the above procedure repeatedly until all the tasks are processed. The controller software also has polling intervals for checking the messages sent by the Arena® supervisor. After receiving each task information from the supervisor, equipment level interactions are then invoked by the controller.

4.3 *Enterprise level simulator and integration*

Figure 11 depicts an information model for enterprise level simulation, which contains order, inventory, logistics, output, shop level simulation and header information. These information can be further decomposed, for example, the inventory information includes the inbound and outbound inventory. Inbound inventory can be further classified into safety stock, capacity and ordering policy information. Logistics information contains factory distance and transport vehicle. Transport vehicle information contains vehicle capacity information. Order information contains order due date information.

Figure 12 shows an exemplary implementation of shop level simulation integration involving one Arena® model (DES), one Simio® model (DES), two AnyLogic® models (one system-dynamics and one agent-based simulator), three process simulators in ESPRIT® software (two snapshots of the process simulator shown in the figure), one robot, one milling machine and one turning machine. Lots of modelling details have been incorporated in each simulation model in this simulation integration platform. The Simio® model is used to simulate processing on wing of an aircraft, and the Arena® model simulates the assembly process at the shop level. The agent-based model simulates the behaviour of workers on the shop floor; the system dynamics model simulates interactions between order cost and finished order price based on the cost information retrieved from the shop floor simulators. The simulation models communicate via SOAL XML-based message through web services depicted in the middle of the figure. This platform has been successfully

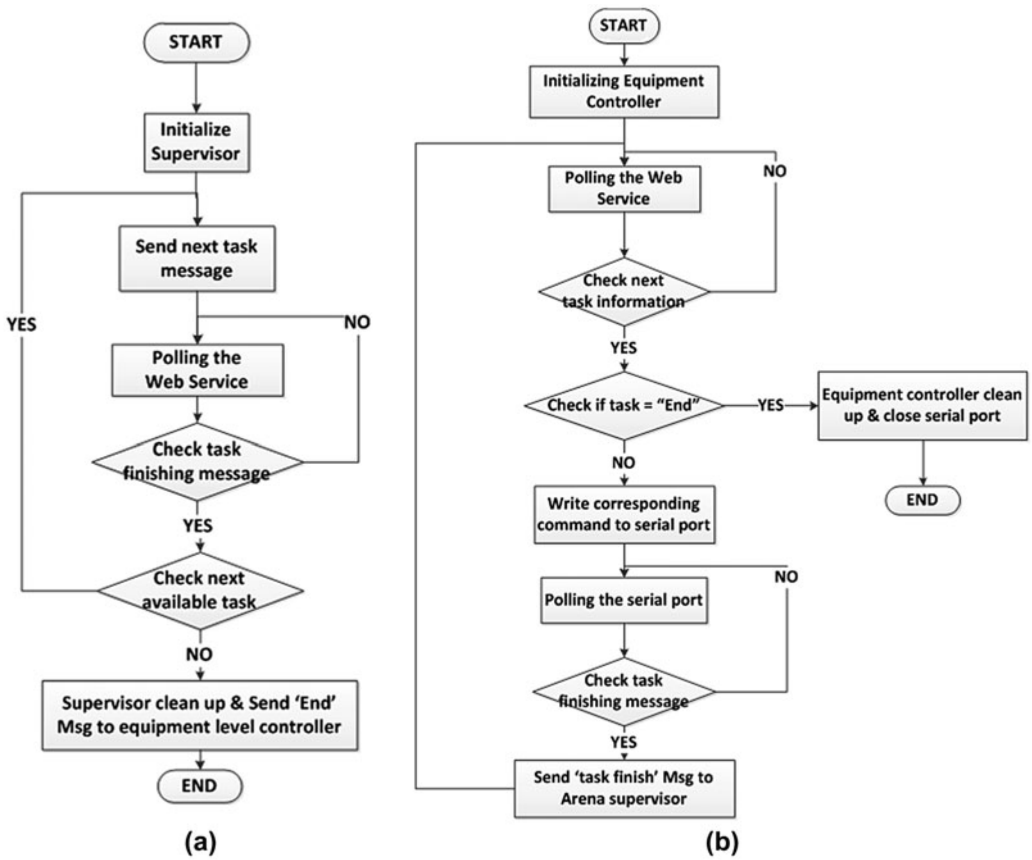

Figure 10. Process flow diagram of SOA for a shop floor control system: (a) Arena® simulation as supervisor, and (b) equipment level controller (Xu 2014).

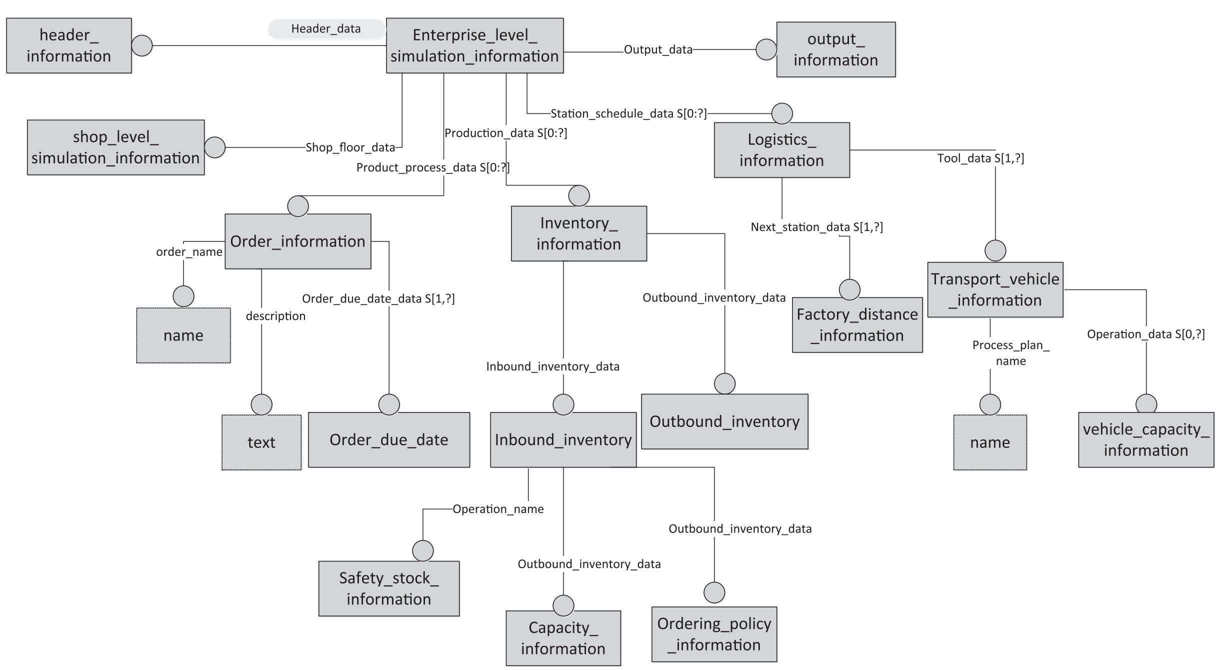

Figure 11. Information model for enterprise level simulation (Son, Jones, and Wysk 2003; Xu 2014).

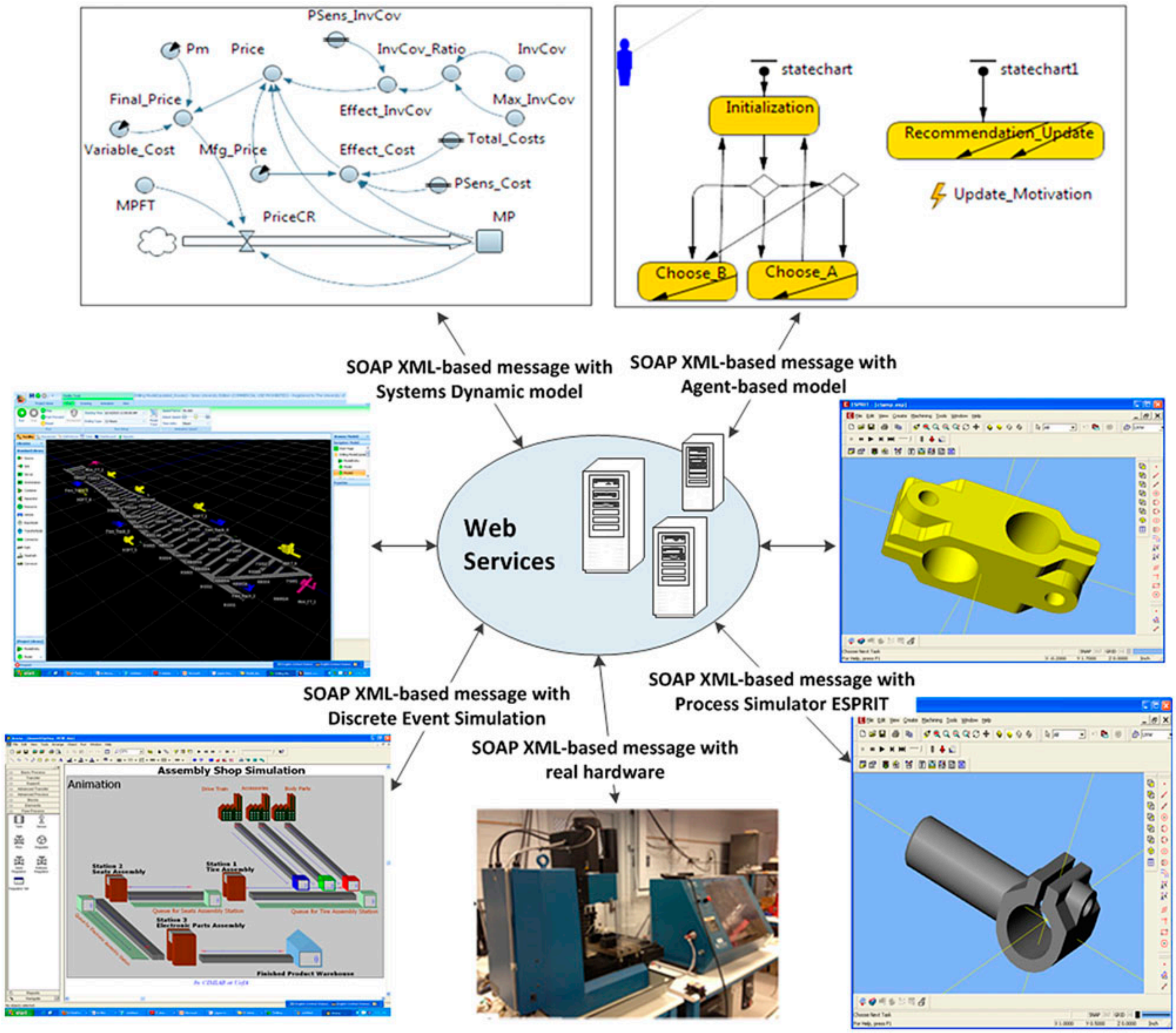

Figure 12. An exemplary implementation of service-oriented simulation integration platform (Xu and Son 2014; Xu 2014).

implemented in the Computer Integrated Manufacturing & Simulation (CIMS) lab at the University of Arizona. During the federation execution, messages exchanged between federates inside each production region may include one or more of the following information: (1) released order information, (2) shop floor status, (3) update-to-date inventory condition, (4) Bill-of-Material (BOM) requirements, (5) process plan information, (6) aggregated time attributes such as part cycle time and resource utilisation associated with the manufacturing and sub-assembly, (7) aggregated cost attributes such as order cost and warehouse inventory cost associated with the manufacturing and sub-assembly, (8) aggregated quality attributes such as part nonconforming rate and resource failure rate associated with the manufacturing and sub-assembly, (9) simulation time advance request and (10) simulation initialisation and termination request. A similar implementation was demonstrated in Xu and Son (2014) as well.

5. Experimental study and results

Most of the experimental results in this section are from a dissertation work (Xu 2014), which is available online. Readers are suggested to read more details about the relevant topics in the dissertation.

In this section, experiments are conducted using the proposed platform to test (1) reduction in estimated variance of part cycle time given variations in order arrival rate, effectiveness of materials processers and material handlers, and (2) computational time performance of Web services. The experiment set-up is discussed in this paragraph. The hierarchical distributed simulation integration platform is run on a homogeneous windows-based platform with eight workstations and a Windows Server. All the workstations and the server are interconnected by a 100 Mbps Ethernet connection. The web services has been developed using Visual C# in the ASP.NET framework and deployed at the Windows Internet Information Services (IIS 6.0). The IIS is hosted on the 32-bit Windows Server® Standard-SP2 (Intel E5530 Quad Core @2.40GHz; 4GB of RAM). Four out of eight workstations (32-bit Windows XP-SP3; Intel P4 1.80GHz, 512MB of RAM) are used for the simulation-based hardware-in-the-loop control of the shop floor machines. The equipment used in the manufacturing test-bed are Intelitek's ProLight™ 1000 Milling Machines, ProLight™ 3000 Turning Machines, SCORBOT-ER V Plus robots and Universal® Laser Machines. The simulation models developed using Arena®, Simio®, AnyLogic® as well as the process simulator ESPRIT® have been deployed on the remaining four workstations (64-bit Windows 7 Professional-SP1; Intel Q9550 Quad Core @2.83GHz, 8GB of RAM).

5.1 *Performance of shop level integration*

In this experiment, two different simulation packages (Arena® and Simio®) are used within a federation. The databases, sample simulation models, application programming interfaces (APIs) as well as client proxies for different simulation federates used are summarised in Table 1. For each federate, interactions with other federates through web services are coded in their respective APIs with the client proxy attached to it. Client proxy is a collection of classes that build and process SOAP XML-based messages on each simulation federate.

A shop performance may become low due to unexpected order, equipment breakdown and material handling uncertainties (Koh and Saad 2002). In this section, we aim to test the performance of shop floor simulation integration platform against some of these factors in addition to material processing uncertainties. We broadly classify the overall effect of these factors in the experiment on the shop floor simulation integration into two different scenarios: (1) low system variations; (2) high system variations. Alternatively, the sources of the system variations applied in this experiment can be resolved into three dimensions: (1) order arrival variation; (2) effectiveness of material processors; (3) effectiveness of material handlers. Order arrival rate is an input to DES (supervisor), in the form of a statistical distribution. Variation in it is implemented using different arrival rate parameters for the same distribution type across different orders. Variation in the material processing is assumed to be due to availability of different tools, machines or human resources for the same task; this variation is large if the observed criteria such as part cycle time or throughput rate differ significantly across different machines or tools. Variation in the material handling is assumed to be due to availability of different material handling methods in the shop floor; this variation is large if the observed criteria such as part cycle time or throughout rate differ significantly across different material handling methods. The parameters for simulating aircraft wing drilling, testing, assembly processes involving wing spars and ribs are listed in Table 2. Under non-simulation integration, the individual simulators are run on initial inputs until a simulation terminating condition is met. Note that no hardware-in-the-loop simulation is involved, as this experimental study focuses on planning.

The statistical distributions for material processing and material handling times follow the uniform and normal distributions, respectively. The parameters of these distributions vary across different processors and handlers, thereby inducing variations in the part cycle times. The enterprise level interactions are incorporated into this experiment even though the focus is on the shop level. Three out of the ten types of information exchanges are considered: BOM requirements, process plan information and aggregated quality attributes (i.e. part non-conformance rate; resource failure rates for processors and handlers only). Part non-conformance and resource failure rates are the same for both low and high system variations. For the convenience of observing part cycle time behaviour as well as changes over the simulation run, the entire simulation time length is partitioned into 10 segments with equal time durations. Box-plots are drawn for each time interval using 20 randomly selected data points. Results are shown in Figures 13 and 14.

Table 1. Information about different federate components (Xu 2014).

Simulation package	Database	API	Client proxy
Arena®	SQL Server	VBA	cls class file
Simio®	Simio® Data Tables	C#	dll class file

Table 2. Parameter setup to estimate a cycle time (Xu 2014).

	Low variations	High variations
Simulation terminating condition	1000 parts	1000 parts
Number of cells in the shop	2	2
Material processors per shop	14	28
Order arrival rate per hour	Expo(5)	Expo(50)
Material handlers per shop	16	32
Number of shops in the factory	4	4
Number of total factory simulators (all discrete-event)	3	3
Number of process simulators in the factory	8	8
Data interaction load per federate under integration	Once in 8 h	Once in 8 h

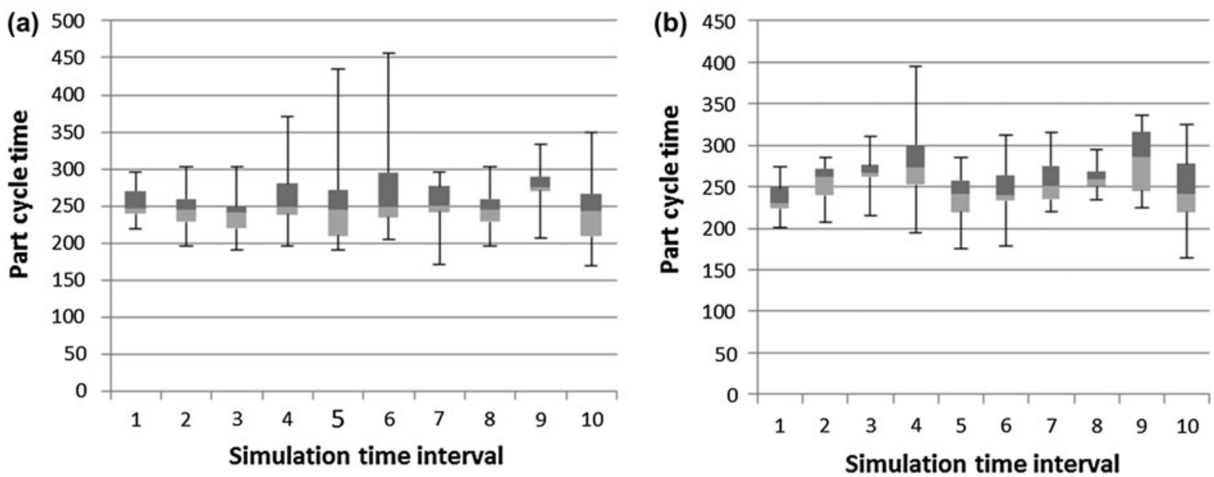

Figure 13. (a) Box-plot for part cycle time under low system variation, non-simulation integration; (b) Box-plot for part cycle time under low system variation, and simulation integration (Xu 2014).

From the box-plots in Figures 13 and 14, we observe that due to shop level simulation integration, the reduction in estimated variance of a part cycle time is higher for the case with high system variations than the case with low system variations. The reduction was 22% under the case with low system variation, and 56% under the case with high system variation. In addition, with low system variations, the simulation integration platform helps to reduce the extreme values of part cycle times in certain cases as observed from the comparison of results under simulation time intervals 5 and 6 (see Figure 13). After inspection, we concluded that these reductions are mainly due to the fact that the integration facilitates the timely detection and resolution of the extreme cases on the shop floor based on the status updates of order/operation and task information. Whenever the drop in the throughput rate is beyond a threshold, alternative process plans which can resolve the delay are retrieved and deployed in the shop floor by the DES. Such alternative process plans may dictate material processor and material handler reallocations or new allocations, whereby extreme values of part cycle time are reduced as shown in Figure 13(b). It is assumed that the alternative process plans do not result in significant increase in the cost. Under high system variation, the cycle times under shop level simulation integration tended to stay between 175 and 360 minutes (see Figure 14(b)) whereas without simulation integration, they range between 80 and 490 minutes (see Figure 14(a)); the variance of the latter case is also higher. In addition, with high system variations without simulation integration (Figure 14(a)), we observed a decrease in the average part cycle time from intervals 2 to 5 and an increase from intervals 5 to 7; but the effects become much smoother with the shop level simulation integration (See Figure 14(b)).

5.2 *Computational time performance of web service*

In this section, two factors that may impact the simulation runtime are selected to test the performance of the web service. They are: (1) interaction load and (2) synchronisation frequency. Four different simulation federations are created,

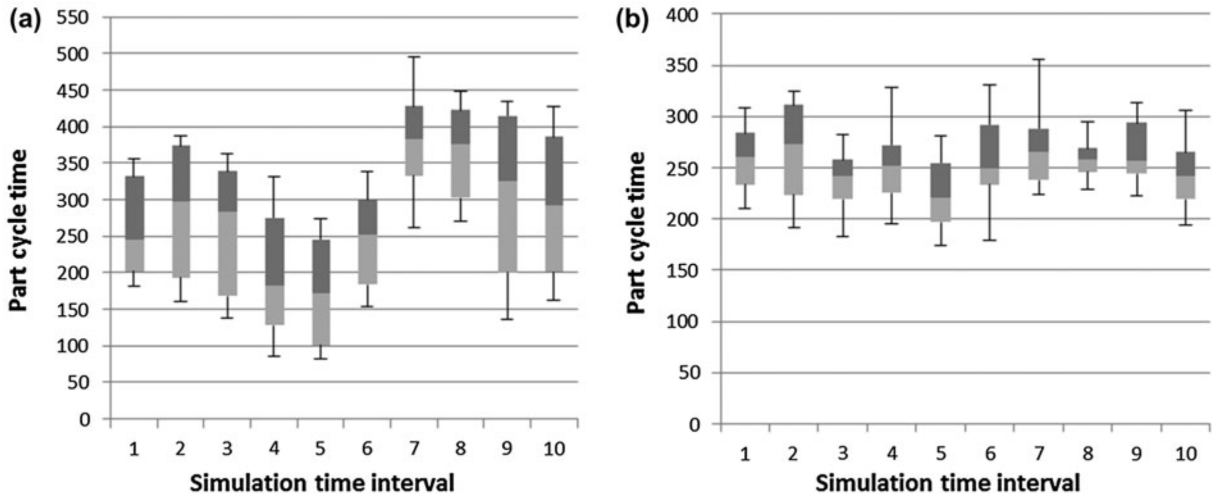

Figure 14. (a) Box-plot for part cycle time under high system variation, non-simulation integration; (b) Box-plot for part cycle time under high system variation, and simulation integration (Xu 2014).

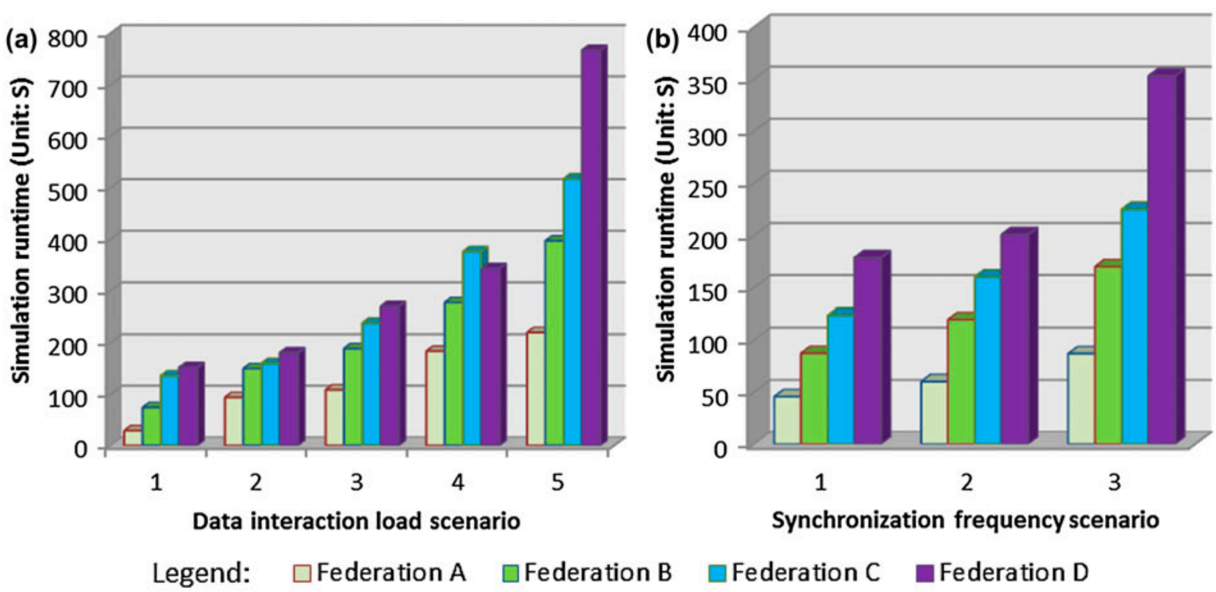

Figure 15. Federation runtime over different interaction loads (a) and synchronisation frequencies (b) (Xu 2014).

each comprising different numbers and types of simulation packages. The simulation federation information is summarised in Table 3. In federations C and D, one system-dynamics-based simulation model implemented in Anylogic® was used; the remaining Anylogic® models were agent-based simulation models. The client proxy files for the Anylogic® models were .jar based and were used to build and process SOAP XML messages for HTTP transport in the web service. The number of process simulators used under each federation configuration is also provided in Table 3. Note that no equipment is involved, as this experiment focuses on planning. All the federations are run under the fast mode without any animations. Before running each simulation federation, all the simulation models were adjusted to run at their respective maximum speeds. For each simulation federation, different data interaction loads and synchronisation frequencies are applied for collecting the distributed simulation runtime information that is summarised in Table 4.

From Figure 15, it is observed that the simulation runtime increases as the interaction load and synchronisation frequency increase. In addition, under the same interaction load and synchronisation frequency, the federation with more federates requires more runtime; this conforms to our intuition. Under the same interaction load, federation with more

Table 3. Parameter set-up for distributed simulation (Xu 2014).

Federation configuration	Number of federates	Composition of federation	Number of process simulators in each federation	Simulation length (h)
Federation A	2	2 Arena® models	2	200
Federation B	4	2 Arena® models, 1 AnyLogic® model, 1 Simio® model	2	200
Federation C	8	4 Arena® models, 2 AnyLogic® models, 2 Simio® models	2	200
Federation D	12	6 Arena® models, 3 AnyLogic® models, 3 Simio® models	2	200

Table 4. Different setups of interaction load and time synchronisation (Xu 2014).

Interaction scenario no.	Data interaction loads with web service	Frequency scenario no.	Time synchronisation frequencies
1	10 interactions per 32 h per federate; Interaction timing follows uniform distribution	1	Once every 3 h, no randomness involved
2	10 interactions per 16 h per federate; Interaction timing follows uniform distribution		
3	10 interactions per 8 h per federate; Interaction timing follows uniform distribution	2	Once every 10 h, no randomness involved
4	10 interactions per 4 h per federate; Interaction timing follows uniform distribution		
5	10 interactions per 2 h per federate; Interaction timing follows uniform distribution	3	Once every 30 h, no randomness involved

federates had an average of 43.84, 24.48 and 11.31% greater simulation runtime when the number of federates increased from 2 to 4, from 4 to 8 and from 8 to 12, respectively. The sample mean runtime is not significantly affected by uncertainty in the timing of the interactions; hence, the confidence intervals have not been plotted. Under the same synchronisation frequency, the federation with more federates had an average of 48.78, 26.47 and 29.11% greater simulation runtime when the number of federates increased from 2 to 4, from 4 to 8, and from 8 to 12, respectively.

6. Conclusions and future work

In this paper, a hierarchical distributed simulation integration platform was proposed for manufacturing planning and control. This platform was based on web services technology that allows integration of various hardware such as milling machine and robot with multi-paradigm, high/low fidelity software components such as DES, process, agent-based and system-dynamics simulators in a realistic and geographically dispersed supply chain environment. Through preliminary results using exemplary models simulating aircraft wing assembly on a shop floor, the benefits of applying horizontal simulation integration among factories at the enterprise level as well as vertical simulation integration between the enterprise level to the shop level and the shop level to the equipment level were demonstrated. At the shop level, estimated variance in part cycle time was reduced despite increasing variations in order arrival rate, effectiveness of material processors and material handlers.

Analysis of the web service performance revealed that under the same interaction load, although the federation with more number of software federates has greater simulation run time, the percentage difference in run time decreases as the number of federates increase. Under the same synchronisation frequency, the percentage difference is smaller between two large federations (4, 8 or 12) than when the number of federates increased from 2 to 4. So, the platform appears to be more beneficial for large supply chains that typically have a greater number of software components. However, it is also observed as expected that within the same federation, the simulation runtime increases whenever the interaction load or synchronisation frequency increases.

Two main extensions are proposed as future works. First, the proposed coherent and comprehensive platform can be tested and extended to various other applications. Second, the distributed-simulation system results can be further investigated by comparing with the centralised system results.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Adams, D., and R. McNamara. 2006. *Service-oriented Architecture and Best Practices*. Accessed March 15, 2015. http://www.omg.org/news/meetings/workshops/soa-bpm-mda-2006/00-T2_Adams.pdf

Alvares, A. J., J. L. N. de Souza Jr., E. L. S. Teixeira, and J. C. E. Ferreira. 2008. "A Methodology for Web-based Manufacturing Management and Control." In *4th IEEE Conference on Automation Science and Engineering*, 668–673. Washington, DC: Key Bridge Marriott.

Arsanjani, A. 2004. "How to Identify, Specify, and Realize Services for your SOA." *Service-oriented Modeling and Architecture*. Accessed March 15, 2015. http://www.ibm.com/developerworks/webservices/library/ws-soa-design1

Bieberstein, N., S. Bose, L. Walker, and A. Lynch. 2005. "Impact of Service-oriented Architecture on Enterprise Systems, Organizational Structures, and Individuals." *IBM Systems Journal* 44 (4): 691–708.

Bozkurt, M., M. Harman, and Y. Hassoun. 2009. "Testing & Verification in Service-oriented Architecture: A Survey." *Software Testing, Verification and Reliability* 23 (4): 261–313.

Chandy, K. M., and J. Misra. 1979. "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs." *IEEE Transactions on Software Engineering* 5 (5): 440–452.

Cho, H., Y. J. Son, and A. Jones. 2006. "Design and Conceptual Development of Shop Floor Controllers through the Manipulation of Process Plans." *International Journal of Computer Integrated Manufacturing* 19 (4): 359–376.

Dahmann, J. S., F. R. Kuhl, and R. Weatherly. 1998. "Standards for Simulation: As Simple as Possible but Not Simpler the High Level Architecture for Simulation." *SIMULATION-Transactions of the Society for Modeling and Simulation International* 71 (6): 378–387.

Ferrari, B. 2011. "Boeing's 787 Program and Another Supply Chain Delay." *Bob Ferrari's Blog on Global Supply Chain Business Process and Technology*. Accessed March 15, 2015. http://www.theferrarigroup.com/supply-chain-matters/2011/07/15/boeings-787-program-and-another-supply-chain-delay/

Fujimoto, R. M. 1998. "Time Management in the High Level Architecture." *SIMULATION-Transactions of the Society for Modeling and Simulation International* 71 (6): 388–400.

Fujimoto, R. M. 2001. "Parallel and Distributed Simulation Systems." In *Proceedings of the 2001 Winter Simulation Conference*, 147–157. Arlington, VA.

Gunsalus, J. 2007. "Boeing Delays 787's First Flight to November–December." *Bloomberg*. Accessed March 15, 2015. http://www.bloomberg.com/apps/news?pid=newsarchive&sid=aUs7S1JX0HuI

Guo, S. 2012. "Simulation Software as a Service and Service-oriented Simulation Experiment." PhD diss., Georgia State University.

Huang, J., M. C. Tung, K. M. Wang, and M. C. Lee. 2005. "Smart Time Management – The Unified Time Synchronization Interface for the Distributed Simulation." *Computer Standards & Interfaces* 27 (2): 149–161.

Jacobi, S., C. Hahn, and D. Raber. 2010. "Integration of Multiagent Systems and Service Oriented Architectures in the Steel Industry." In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Toronto, 479–482.

Karnouskos, S., O. Baecker, L. M. S. de Souza, and P. Spieb. 2007. "Integration of SOA-ready Networked Embedded Devices in Enterprise Systems via a Cross-Layered Web Service Infrastructure." In *IEEE Conference on Emerging Technologies and Factory Automation*, Patras, 293–300.

Koh, S. C. L., and S. M. Saad. 2002. "Development of a Business Model for Diagnosing Uncertainty in ERP Environments." *International Journal of Production Research* 40 (13): 3015–3039.

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

Lee, S., Y. J. Son, and R. A. Wysk. 2007. "Simulation-based Planning and Control: From Shop Floor to Top Floor." *Journal of Manufacturing Systems* 26 (2): 85–98.

Lendermann, P., N. Julka, B. P. Gan, D. Chen, L. F. McGinnis, and J. P. McGinnis. 2003. "Distributed Supply Chain Simulation as a Decision Support Tool for the Semiconductor Industry." *SIMULATION: Transactions of the Society for Modeling and Simulation* 79 (3): 126–138.

Lutz, R., R. Scrudder, and J. Graffagnini. 1998. "High Level Architecture Object Model Development and Supporting Tools." *SIMULATION-Transactions of the Society for Modeling and Simulation International* 71 (6): 401–409.

Mendes, J. M., P. Leitão, A. W. Colombo, and F. Restivo. 2012. "High-level Petri Nets for the Process Description and Control in Service-oriented Manufacturing Systems." *International Journal of Production Research* 50 (6): 1650–1665.

Misra, J. 1986. "Distributed Discrete-event Simulation." Computer Surveys 18 (1): 39–65.

Misra, S., J. Venkateswaran, and Y. J. Son. 2003. "Framework for Adaptive Time Synchronization Method for Integration of Distributed, Heterogeneous, Supply Chain Simulations." In *Proceedings of the 2003 American Society of Engineering Management Conference*, St Louis, MO.

Ollinger¸ L., J. Schlick, and S. Hodek. 2011. Leveraging the Agility of Manufacturing Chains by Combining Process-oriented Production Planning and Service-oriented Manufacturing Automation. In *18th International Federation of Automatic Control (IFAC) World Congress*, 5231–5236. Milano.

Ostrower, J. 2011. "Boeing Holding 787 Line for a Month, to Update 2011 Delivery Guidance." *Flightglobal Blog*. Accessed March 15, 2015. http://www.flightglobal.com/blogs/flightblogger/2011/07/boeing_holding_787_line_for_a/

Papazoglou, M. P., and W.-J. van den Heuvel. 2007. "Service Oriented Architectures: Approaches, Technologies and Research Issues." *The VLDB Journal* 16: 389–415.

Rathore, A., B. Balaraman, X. Zhao, J. Venkateswaran, Y. J. Son, and R. A. Wysk. 2005. "Development and Benchmarking of an Epoch Time Synchronization Method for Distributed Simulation." *Journal of Manufacturing Systems* 24 (2): 69–78.

Ray, S. 2008. "Boeing Reviews Dreamliner Schedule for More Delays." *Bloomberg*. Accessed March 15, 2011. http://www.bloomberg.com/apps/news?pid=newsarchive&sid=aSpIiLLg_Uj8&refer=home

Riley, G. F., M. H. Ammar, R. M. Fujimoto, A. Park, K. Perumalla, and D. Xu. 2004. "A Federated Approach to Distributed Network Simulation." *ACM Transactions on Modeling and Computer Simulation* 14 (2): 116–148.

Schmidt, M.-T., B. Hutchison, P. Lambros, and R. Phippen. 2005. "The Enterprise Service Bus: Making Service-oriented Architecture Real." *IBM Systems Journal* 44 (4): 781–797.

Smith, J. S., R. A. Wysk, D. T. Sturrok, S. E. Ramaswamy, G. D. Smith, and S. B. Joshi. 1994. "Discrete Event Simulation for Shop Floor Control." In Proceedings of the 1994 Winter Simulation Conference, 962–969. Lake Buena Vista, FL.

Son, Y. J., S. B. Joshi, R. A. Wysk, and J. S. Smith. 2002. "Simulation-based Shop Floor Control." *Journal of Manufacturing Systems* 21 (5): 380–394.

Son, Y. J., A. T. Jones, and R. A. Wysk. 2003. "Component Based Simulation Modeling from Neutral Component Libraries." *Computers & Industrial Engineering* 45 (1): 141–165.

Steele, J., Y. J. Son, and R. A. Wysk. 2001. "Resource Modeling for the Integration of the Manufacturing Enterprise." *Journal of Manufacturing Systems* 19 (6): 407–427.

Strassburger, S., T. Schulze, and R. M. Fujimoto. 2008. "Future Trends in Distributed Simulation and Distributed Virtual Environments: Results of a Peer Study." In *Proceedings of the 2008 Winter Simulation Conference*, 777–785. Miami, FL.

Tao, F., Y. Cheng, L. D. Xu, L. Zhang, and B. H. Li. 2014. "CCIoT-CMfg: Cloud Computing and Internet of Things-based Cloud Manufacturing Service System." *IEEE Transactions on Industrial Informatics* 10 (2): 1435–1442.

Taylor, S. J. E., R. Sudra, T. Janahan, G. Tan, and J. Ladbrook. 2002. "GRIDS-SCF: An Infrastructure for Distributed Supply Chain Simulation." *SIMULATION-Transactions of the Society for Modeling and Simulation International* 78 (5): 312–320.

Terzi, S., and S. Cavalieri. 2004. "Simulation in the Supply Chain Context: A Survey." *Computers in Industry* 53 (1): 3–16.

Valilai, O. F., and M. Houshmand. 2013. "A Collaborative and Integrated Platform to Support Distributed Manufacturing System Using a Service-oriented Approach Based on Cloud Computing Paradigm." *Robotics and Computer-Integrated Manufacturing* 29 (1): 110–127.

van der Zee, D. J., and J. van der Vorst. 2005. "A Modeling Framework for Supply Chain Simulation: Opportunities for Improved Decision Making." *Decision Sciences* 36 (1): 65–95.

Venkateswaran, J., and Y. Son. 2004. "Design and Development of a Prototype Distributed Simulation for Evaluation of Supply Chains." *International Journal of Industrial Engineering* 11 (2): 151–159.

Venkateswaran, J., and Y. J. Son. 2005. "Hybrid System Dynamic–Discrete Event Simulation-based Architecture for Hierarchical Production Planning." *International Journal of Production Research* 43 (20): 4397–4429.

Wysk, R. A., B. A. Peters, and J. S. Smith. 1995. "A Formal Process Planning Schema for Shop Floor Control." *Engineering Design and Automation Journal* 1: 3–19.

Xu, D. 2014. "An Integrated Simulation, Learning and Game-theoretic Framework for Supply Chain Competition." PhD diss., Department of Systems and Industrial Engineering, University of Arizona.

Xu, D., and Y. J. Son. 2014. "Production Planning and Control via Service-oriented Simulation Integration Platform." In Proceedings of the 2014 Industrial & Systems Engineering Research Conference, Montreal, 351–360.

Xu, D., S. S. Nageshwaraniyer, Y. J. Son, and S. Song. 2011. "Simulation-based Assessment of Change Propagation Effect in an Aircraft Design Process." In Proceedings of the 2011 Winter Simulation Conference, Phoenix, AZ, 1715–1726.

Accessed February 10, 2012. http://en.wikipedia.org/wiki/High_Level_Architecture_(simulation)

Accessed March 15, 2015. http://en.wikipedia.org/wiki/Web_service

Zhang, Y., X. Xu, Y. Liu, and X. Bai. 2012. "Service-oriented, Cross-platform and High-level Machining Simulation." *International Journal of Computer Integrated Manufacturing* 25 (3): 280–295.

Zhong, R. Y., Z. Li, L. Y. Pang, Y. Pan, T. Qu, and G. Q. Huang. 2013. "RFID-enabled Real-time Advanced Planning and Scheduling Shell for Production Decision Making." *International Journal of Computer Integrated Manufacturing* 26 (7): 649–662.