

Service-oriented simulation framework: An overview and unifying methodology

Wenguang Wang, Weiping Wang, Yifan Zhu and Qun Li

Abstract

With the prevalence of net-centric environments, Service-Oriented Architectures (SOAs) have emerged as a paradigm that greatly impacts the modeling and simulation (M&S) community. This paper has two interrelated goals. The first is to give a comprehensive review of various service-oriented simulation frameworks to help researchers select the appropriate one for their specific purpose. The second goal is to combine the common features derived from the review into one unifying framework that can describe and prescribe various specific approaches. The focus of this paper is on the common functionalities of service-oriented simulations reflected in the review and unifying framework. In particular, we emphasize the way SOAs and M&S are combined, and the interoperability and composability challenges of distributed M&S services. We describe some fundamental concepts first. Then we present a comprehensive survey of several classical frameworks, including formalism-based, model-driven, interoperability protocol based, eXtensible Modeling and Simulation Framework (XMSF), Open Grid Services Architecture (OGSA) based, and ontology-driven frameworks. Based on the review, we propose a novel three-dimensional reference model that can unify the ad hoc approaches into a common framework. The model can be used as a guideline or an analytic means to find potential and possible future directions. In particular, the model inspects the crossover between the disciplines of M&S, service-orientation, and software/systems engineering. Based on the model, we present a detailed comparison of the reviewed frameworks. Finally the significance of the paper is discussed and future directions are recommended.

Keywords

Modeling, simulation, Service-Oriented Architecture (SOA), software engineering, systems engineering, Web services, composability, interoperability

1. Introduction

With the prevalence of net-centric environments,¹ it is increasingly important for the modeling and simulation (M&S) community to offer agile on-demand capabilities by providing, reusing, and composing heterogeneous resources. Some limitations of stand-alone and distributed simulation frameworks have been revealed, such as accessibility, interoperability, composability, extensibility, agility, and reusability. These frameworks need new methodologies and techniques to embrace net-centric environments with properties of distribution, sharing, and collaboration.

Software techniques and methodology, in particular Service-Oriented Architectures (SOA),^{2,3} provide such an opportunity. The SOA separates the concerns of the stakeholders. A service provider can encapsulate various resources or capabilities into services, and then

publish implementation-independent service descriptions via a service broker. A service requestor can ascertain the required services from the broker and combine these with other services. The SOA comprises a set of international Web standards that enhance the accessibility and interoperability of heterogeneous resources. Information enterprises⁴ and others can benefit from service-oriented approaches to improve the agility and

Department of Systems Engineering, College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China.

Corresponding author:

Wenguang Wang, Department of Systems Engineering, College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China
Email: wgwangnuds@gmail.com

flexibility in the event of unanticipated changes in the requirements. Therefore service-orientation is emerging as a new paradigm in systems analysis and software development. As a result, various new terms have been defined such as service-oriented science,⁵ computing,⁶ modeling,⁷ simulation,⁸⁻¹¹ system engineering,¹² software engineering,^{13,14} and high level architecture (HLA).¹⁵ Major computer corporations, some international standards organizations, and many programs, such as the Department of Defense (DoD) net-centric enterprise services,⁴ net-centric services strategy,¹⁶ and the DoD architecture framework¹⁷ amongst others, are embracing the new paradigm.

In the M&S community, the use of SOAs to extend the capabilities of the M&S framework has attracted increasing attention.¹⁸ Service-oriented approaches can benefit models and simulations by addressing properties of accessibility, reusability, composability, and so on. They can also change the way M&S are deployed and used,¹⁹ and facilitate simulation on demand in a net-centric environment.²⁰ From the perspective of techniques and tools for Web-based simulation, Byrne et al.²¹ conducted a review of the pros and cons, a classification of different sub- and related-areas, enabling technologies, and the evolution of Web-based simulation. From the perspective of methodologies and formalisms for service-oriented simulation, various service-oriented simulation frameworks have been proposed or implemented by different institutes using different formalisms and techniques. These include formalism-based,^{22,23} model-driven,²⁴ interoperability protocol based,¹⁵ Open Grid Services Architecture (OGSA) based,²⁵ and ontology driven^{26,27} frameworks, as well as the Extensible Modeling and Simulation Framework (XMSF).²⁸ All of these reflect the net-centric objectives and the way M&S and SOA are combined according to different aspects.

However, two key challenges still need to be addressed in the research of service-oriented simulation. First, each framework proposed thus far has its unique properties as well as pros and cons. It is thus necessary to undertake a comprehensive review of related concepts, issues, techniques, and the state-of-the-art of various service-oriented simulation approaches. Such a review may facilitate the classification, evaluation, selection, implementation, extension, and application of the reviewed or future frameworks. Second, the frameworks proposed thus far generally focus on specific domains or applications. They are capable of addressing different issues within service-oriented simulation from different viewpoints. However, there has been little work on the common functionalities and totality of research issues reflected in various specific frameworks. Such an investigation may lead to a general (or high level) systematic methodology for service-oriented simulation derived

from the state-of-the-art. The methodology may facilitate describing, analyzing, developing, and addressing issues in service-oriented simulation in a systematic way. In addition, such a methodology has both inductive and deductive uses, by which researchers can explore, select, and synthesize specific issues, formalisms, and approaches based on their requirements.

Considering the above, this paper has two interrelated goals. The first is to undertake a comprehensive review of various service-oriented simulation approaches to help researchers select the appropriate one for their specific purpose. The second goal is to combine the common features derived from the state-of-the-art into one unifying framework that can describe and prescribe various specific approaches. The reviewed approaches may justify the unifying framework in return. The message and focus of this paper is the common functionalities (i.e. the three-dimensional aspects of service-oriented simulation) reflected by the reviewed approaches and the unifying framework. In particular, we pay more attention to the way that SOAs and M&S are combined, and the interoperability and composability challenges of distributed M&S services.

The rest of the paper is organized as follows. In Section 2, some fundamental concepts of service-oriented simulation are discussed. In Section 3, we give a comprehensive survey of various service-oriented simulation approaches. Deriving from the review, we propose a novel three-dimensional reference model in Section 4 as a unifying framework and methodology for service-oriented simulation. In Section 5 we use the unifying frame to describe, compare, and prescribe the reviewed approaches in detail from the viewpoint of one, two, and three dimensions demanded by the three-dimensional (3D) model. In Section 6 the significance of the paper is discussed and future directions are recommended.

2. Concept exploration

As a basis for later investigation, we first explore related concepts of service-oriented simulation.

2.1. Services

Services have different implications in different contexts. Quartel et al.,²⁹ Balin and Giard³⁰ summarized the definitions of services and their properties from the process, interaction, capability, and operation point of view, amongst others. Two prevailing definitions have been given by the World Wide Web Consortium (W3C) and the DoD. Within the domain of IT, the W3C defines a service as ‘an abstract resource that represents a

capability of performing tasks that form a coherent functionality from the point of view of providers entities and requesters entities'.³¹ On the other hand, in the defense community, the DoD defines a service as 'a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description'.¹⁶ Based on the definitions of a 'service' given above and by others, much attention is paid to capability, utility, interface, and functionality aspects, whereas implementation details are generally hidden. The W3C's definition focuses on the IT resources, while the DoD's provides a detailed mechanism with essential elements and constraints to access various distributed capabilities. Additionally, the DoD is the key initiator and driver of the M&S and net-centric strategy. Hence, we prefer the DoD's definition in this paper.

Given these characteristics, services have different taxonomies according to the types of capability, carrier, presentation, application scope, context, and so on. For example, the US net-centric services strategy classified services as Core Enterprise Services (CES) and Communities of Interest (COIs) services.¹⁶ Tolk et al.³² divided the services that access the Common Reference Model (CRM) into atomic, composite, aggregate, and data mediation services from the perspective of model-based data engineering. Suzić et al.³³ proposed a services taxonomy of Operational, System Management, Messaging, Registration and Discovery, Mediation, Collaboration, Information Assurance and Security, Storage, and Application Services in their core technical framework.

2.2. Simulation services

Similarly, as a special kind of service, simulation services have different implications in different contexts. For example, in the context of a HLA, simulation services may refer to runtime infrastructure (RTI) services for models, such as time management and object management.

Taking the Web as an implementation platform, Zhang et al.^{26,34} defined *simulation services* as simulation components encapsulated with certain simulation applications or model logics, which have certain functions and are embodied as state-persistent Web services. The information and semantics of simulation services are described by Web service standards. The communication and interoperation among services are enabled by standard Web service protocols. Simulation services satisfy user's requirements through cooperation of all involved services.

In this paper, we incorporate the general M&S capabilities into the definition of services. Therefore, we

define a *general simulation service* from a capability perspective as follows:

A 'simulation service' refers to one or more capabilities implicated in abstract (i.e. conceptual) or implemented by concrete (i.e. implementation related) M&S artifacts that can be accessed by consumers.

Meanwhile, we regard Zhang's implementation-related definition^{26,34} in the narrow sense. The definitions of simulation services in both general and narrow senses are given to facilitate the service-oriented concept, and the analysis, design, and implementation thereof.

2.3. Service-oriented simulation

To the best of our knowledge, the idea of service-oriented simulation was first implicated in the XMSF project.^{8,9} Thereafter the concept of service-oriented simulation was explicitly proposed by Gustavsson et al.^{10,11} but from the viewpoints of the Swedish Armed Force Enterprise Architecture Services, simulation, and software engineering. As such, it has not evolved into the general purpose service-oriented simulation concept as a successor to object-oriented simulation.³⁵ Referring to the object-oriented, process-oriented, and event-oriented simulation concepts laid out by the DoD,³⁶ we define *service-oriented simulation* as:

A simulation using a service-oriented paradigm in which the service and its capability are considered more important than the object, process, or outcome. Service-oriented simulation focuses on the modeling, description, publication, discovery, composition, orchestration, simulation, etc. in the lifecycle of services or simulation services. For example, in service-oriented war game simulation, an observation service pays more attention to observation capability than concrete processes or objects (such as human vision, telescope, and radar).

There are three distinct yet related concepts about service-oriented simulation. The first is 'Service-based simulation', which implies using only the basic SOA language/platform independent concepts/properties (corresponding to core issues, e.g. service provider and requestor), and not the full potential of SOA (indirect addressing, broker, composition, etc.). The second is 'Service-oriented simulation', which uses the full potential of SOA especially broker and composition. (This corresponds to core and supporting issues.) The third is 'Service-oriented simulation engineering', which emphasizes the use of engineering principles or approaches in the service-oriented simulation concept. (This corresponds to the general service-oriented simulation.) In this work, we only use the general 'service-oriented simulation' concept to represent the above three detailed definitions.

Service-oriented simulation has two research directions.¹⁸ One is the application of M&S to SOA, e.g. using M&S techniques to address the analysis, design, evaluation, and testing problems in service-oriented systems. The other is the application of SOA to M&S, e.g. using the service-oriented paradigm to extend the capability of M&S techniques and frameworks. In this work, we pay more attention to the latter direction.

The tasks in service-oriented simulation include: (i) the identification of applicable services (based on conceptual views of the task to be supported and the services that can provide this task); (ii) selection of the best alternative (if more than one service is applicable, based on concept and implementation); (iii) the composition of these services (to create a gapless seamless match to the proposed conceptual model); and (iv) the orchestration of the execution. Wallace et al.³⁷ proposed an eight-step composition process for object models, which can be adapted to the services composition process.

2.4. Service-oriented modeling and simulation framework

To define a *service-oriented modeling and simulation framework*, the definition of architecture in software-intensive systems is reviewed:³⁸ ‘*An architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution*’. Regarding architecture style, two kinds prevail. One is component oriented that emphasizes components and the encapsulation of attributes and functions (e.g. object-oriented systems). The other is connector/relationship oriented that emphasizes the interface, communication protocols, and composition between the components (e.g. service-oriented systems).

In the M&S community, Zeigler et al.³⁹ defines a modeling and simulation framework as: ‘*a framework that defines entities and their relationships that are central to the M&S enterprise. The basic entities of the framework are source system, model, simulator, and experimental frame. The basic interrelationships among entities are the modeling and the simulation relationships*’.

To cover both the software and M&S characteristics, we define a *service-oriented modeling and simulation framework* as:

A service-oriented modeling and simulation framework is the fundamental organization of a service-oriented simulation system that represents its components (services/simulation services), the components’ relationships to one another, and to the environment in the system lifecycle of modeling, description,

publication, composition, orchestration, simulation, etc., and the principles that guide its design and evolution.

Note that models are one of the key components for composable M&S services. This is also reflected in the DoD net-centric data and service strategies. Regarding services, we focus on their conceptual level for composability and their interface level for interoperability. Because services concentrate more on the capabilities and interfaces than the inner implementation, we address the issues of a service-oriented simulation framework without emphasizing the implementation of the services. As such we do not care how services are built by various programming languages, middleware, platforms, and so on.

2.5. An example of a service-oriented simulation

Here we use an assumed war game simulation as an example to show the related concepts and processes of service-oriented simulation. Recommended major steps and activities are illustrated in Figure 1.

1. A military enterprise, e.g. the DoD, intends utilizing M&S techniques to evaluate the effectiveness of the missile defense system in a net-centric environment. This may provide a reference to the design and deployment of real systems. – Define simulation objectives.
2. The war game environment consists of a certain threat (i.e. an attacking missile), observation services (i.e. satellite and radar), orientation and decision services (i.e. Command and Control system, C2), and a defense service (i.e. intercepting missile). The DoD wants to use a certain simulation runtime infrastructure to connect all the model services and execute them. Instead of developing a new system from scratch, the DoD wants to reuse a legacy military service/model/simulator. The DoD also wants to integrate heterogeneous, distributed components. – Perform conceptual analysis, scenario or experimental frame development, identify conceptual services.
3. The DoD searches the service broker for required model services and the ontology library for exact terms of Quality of Service (QoS) parameters (e.g. the azimuth angle of the radar). The service broker supports a vague search via a Web browser. – Design simulation environment. User searches the broker for required services.
4. A list of candidate services is acquired. The DoD selects the exact matched services according to other QoS information (e.g. price, reliability, access scheme). The DoD obtains the service description of various completely matched services

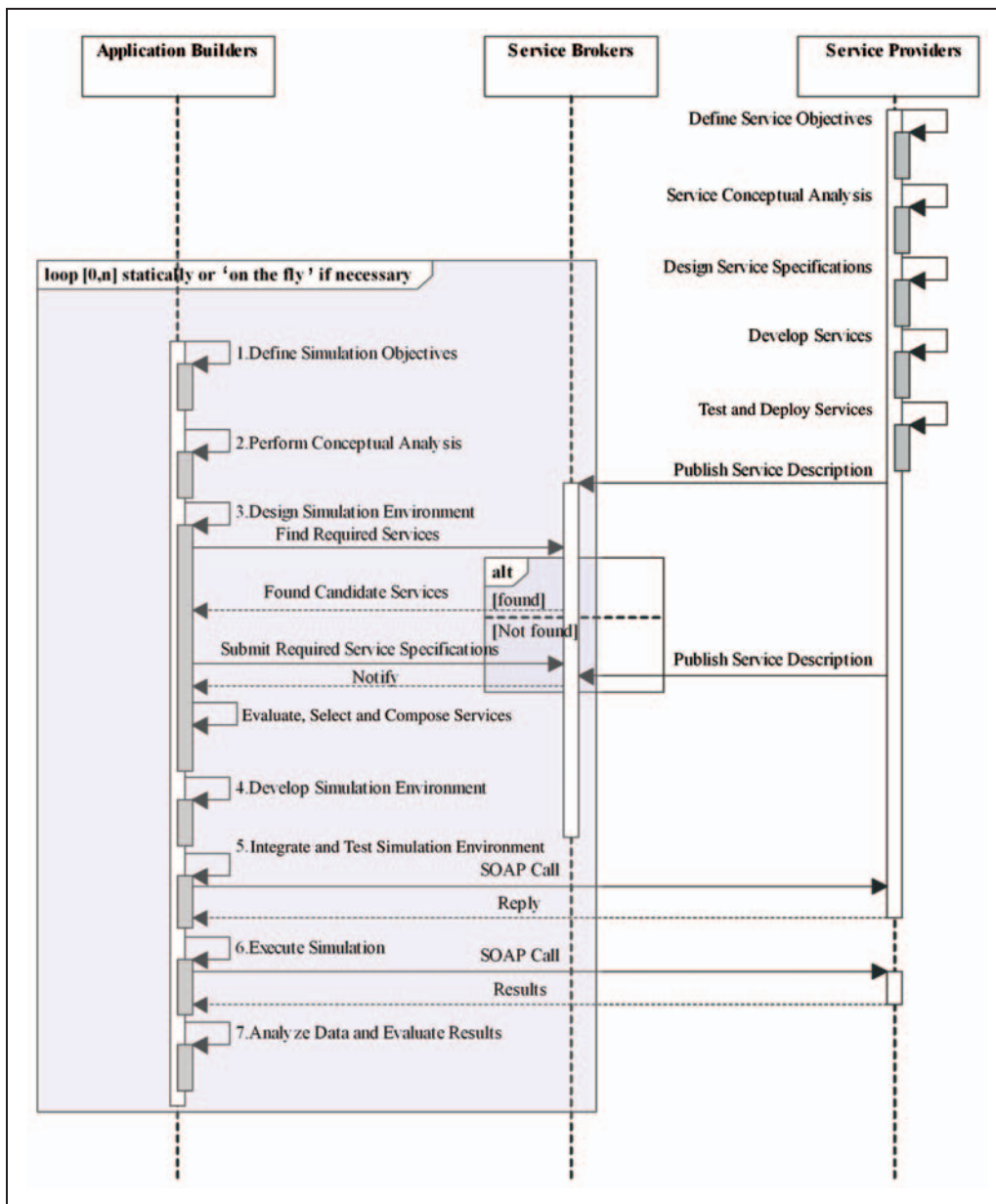


Figure 1. Recommended practice for service-oriented simulation engineering and execution process.

- (e.g. missile, satellite, and C2). – Get service description with URL.
- 5. The candidate radar services are not identically matched (e.g. the detect range of the radar is not satisfactory.) In addition, no defense missile services have been found. – No required service.
- 6. The DoD writes the specifications for the required services for radar and a defense missile. – Required service specification.
- 7. The DoD submits the specifications to the service broker or other directories.
- 8. One radar manufacturer and one defense missile manufacturer respond to the ads. They believe

- they can provide the required services. – Service provider.
- 9. The radar manufacturer improves the legacy radar service and provides the required service description to the broker. The defense missile manufacturer produces and submits its service description to the broker. – Each provider registers its service with the broker. Develop member applications.
- 10. The broker informs the DoD, which has now acquired all the required model services. – Get all the implementation-related model services.
- 11. The DoD searches for simulation infrastructure services. It adopts the HLA Evolved Web service

- RTI. – Get the implementation-related simulation service.
12. The DoD aligns the conceptual/modeling properties, such as objectives, assumptions, and constraints for the services/models. – Composition of conceptual services.
 13. The DoD develops simulation data exchange model and establish simulation environment agreements. – Develop simulation environment.
 14. The DoD composes and integrates all the services in a workflow manner. – Service orchestration or static composition.
 15. The DoD confirms the availability of all the services. The services authenticate the user (i.e. the DoD). – Integrate and test simulation environment.
 16. The simulation is tested over the Web. Messaging through HTTP, SOAP, etc.
 17. The DoD obtains the results, which show that the range of the defense missile impacted on the defense effectiveness.
 18. The DoD searches for defense missile services with a higher range.
 19. The required service is found. Meanwhile, the DoD finds a better radar service with higher reliability and lower price. – QoS management.
 20. The old defense missile and radar services are replaced before execution or ‘on the fly’ by some service agents. – Replacement, choreography, or dynamic composition of services.
 21. The simulation is executed over the Web. – Execute simulation.
 22. The DoD obtains the required results. The selection and deployment of proper equipment can be recommended based on the simulation results. – Analyze data and evaluate results.
 23. The DoD pays the appropriate fees for the delivered services.
 24. The DoD saves the services contact/description for future use.

We obtain the following observations and make further explanations to the above example and Figure 1.

- (i) *The persistence of services.* This is a typical example of military engagement. In particular, the Observe-Orient-Decide-Act (OODA)⁴⁰ model depicted by this example is widely adopted in both military and other domains. Even in wars in the old Roman Empire, threat, observation, orientation/decision, and act services were present (e.g. enemy, human vision, leaders, and arrows, respectively). The differences lie in the parameters or QoS. Services survive much longer than their implementations.
- (ii) *Further explanations for the steps.* Detailed activities in each step can refer to the Distributed Simulation Engineering and Execution Process (DSEEP).⁴¹ Three patterns of interactions (recruitment, recommendation, and notification) between service requestors and providers via brokers can refer to the work of Yilmaz and Paspuleti.⁴² Detailed evaluation, selection, composition, and alignment processes of service components can refer to.^{37,43–45}
- (iii) *The advantages of service-oriented simulation.* From the application builders’ (service requestors’) perspective, they have similar major steps as those of the traditional distributed simulation like the HLA. However, the significant advantage lies in the possibility to find, select, and compose solutions out of provided services ‘on the fly’ based on the right meta-data (service descriptions).^{18,24,46} Hence it is possible for service-oriented simulations to improve the agility and flexibility in the event of unanticipated changes in the requirements.⁴⁷ In addition, the SOA has the nature of loose coupling. It separates the concerns of consumers and providers. It also separates service descriptions from their implementations (platforms, languages, etc.). The brokers provide an open market for the publication, discovery, and composition of various services. The SOA is implemented by open Web standards that can facilitate a variety of communities to provide their information assets as services. Therefore, it is possible for users to obtain the required services from other broad communities (e.g. C4I) besides M&S⁴⁸ when and where they are needed. Other advantages of service-oriented simulations can refer to the authors’ previous work.¹⁵
- (iv) *Regarding semi-automated or automated orchestration and composition of services.* The idea of service-oriented simulation is to move toward semi-automated orchestration (with composition being the ultimate goal, but this is much harder to reach). The classical service-oriented simulation frameworks, as well as our assumed example, revealed the basic advantages mentioned in (iii), but still in an infant stage toward the advanced goal. Machine-readable annotations for services, as requested by Tolk et al.³² are one of the needed improvements. Meta-data, semantics, and ontologies of services can facilitate intelligent agents to orchestrate and compose services semi-automatically or automatically in the future.^{49,50}
- (v) *The relationship to reviewed or proposed frameworks in this paper.* The purpose of the example is to give general readers a clear and understandable background to the related concepts and processes.

The example reveals the M&S, service-orientation, and engineering aspects of service-oriented simulation. However, as it is part of the background section, the example does not intend to validate a specific approach or the 3D unifying framework. Validation of specific examples belongs in a discussion of specific approaches.^{22,24}

3. An overview of classical service-oriented simulation frameworks

Based on related concepts, this section describes the state-of-the-art of several classical service-oriented simulation frameworks. These combine M&S and SOAs to address the issues of service-oriented simulation in different ways. Each framework has its own advantages and limitations.

3.1. Formalism-based framework

This kind of framework depends on certain simulation formalisms in a theoretical or mathematical way. Discrete Event System Specification (DEVS) is a typical example that includes the following extensions.

3.1.1. DEVS unified process framework (DUNIP). The DEVS Unified Process framework (DUNIP)²² was proposed by Mittal for the integrated development and testing of service-oriented architectures.

From an M&S perspective, using DEVS as a unified model specification, the automated generation of DEVS models from a number of different formalisms such as state-based, rule-based, BPMN/BPEL-based, and DoDAF-based was investigated. The DEVS/SOA²⁰ depicted in Figure 2 was proposed as a simulation service platform to address simulator compatibility issues such as DEVS/C++, DEVSJAVA, DEVS/RMI, and so on. To increase interoperability, the simulation processes are totally transparent with respect to model execution over the net-centric infrastructure. Users can execute models over the Internet using Web services and SOA protocols. *From the viewpoint of composability and interoperability*, the composition and execution of models conform to the System Entity Structure (SES), modular, hierarchical DEVS specification, and DEVS simulation protocols³⁹.

From a service-orientation viewpoint, DEVS models are regarded as resources, while simulators are regarded as Web services. The DEVS Modeling Language (DEVSML)⁵¹ was proposed to present DEVS models in XML format. All DEVS models are independent of the implementation platform, thus increasing the reusability, composability, and extensibility thereof.

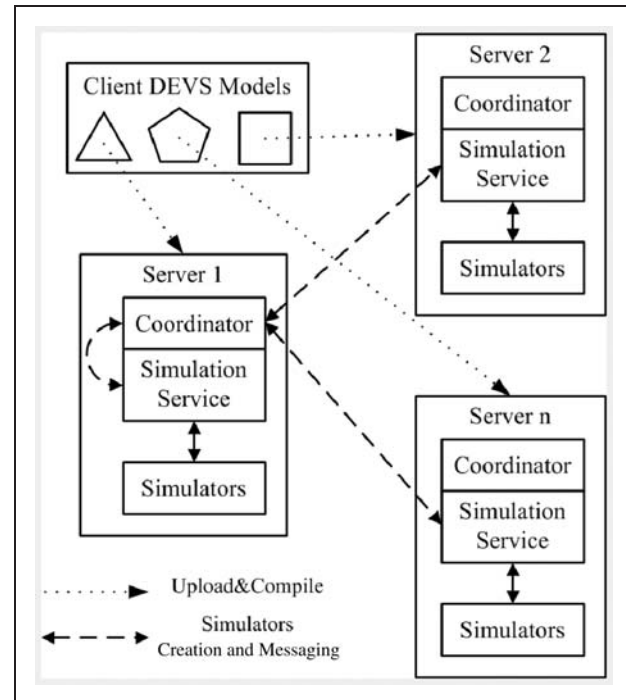


Figure 2. DEVS/SOA architecture.

The hierarchical architecture of DEVSML was reported by Mittal et al.⁵¹ An approach using an abstract wrapper that automatically generates the DEVS Web service from a WSDL interface was presented by Mittal et al.⁵² The abstraction mechanism of a coupled model as an atomic model with a DEVS state machine and its implementation, i.e. the adapter Digraph2Atomic, was reported by Mittal.²² A coupled model can be executed like an atomic model. Hence, simulator services are sufficient to execute DEVS models over net-centric environments without coordinator services. The early version of the DEVS/SOA used a centralized communication mechanism with a central coordinator. The latest version utilizes direct and real-time communication between services.⁵³

From a software/systems engineering viewpoint, the lifecycle of bifurcated model-continuity methodology was proposed in DUNIP to unify the concepts of model-continuity and the M&S framework. The complete process of DUNIP starts from the automated generation of DEVS models from various requirements specifications. Then, the DEVS models are transformed to platform-independent XML format using DEVSML. The DEVS/SOA simulation platform is used to deploy, simulate DEVS models, and collect output. The architecture and processes of DUNIP were shown by Mittal and Zeigler.⁵⁴

DUNIP has been partly applied in several projects,²² e.g. the Joint Close Air Support (JCAS) Model, the

DoDAF-based Activity Scenario, the Link-16 ATC-Gen Project at JITC, and the GENETSCOPE Project at JITC. The DEVS/SOA and DUNIP are important infrastructures for net-centric information exchange and systems of system interoperations.^{55,56}

3.1.2. DEVS simulation framework for service-oriented computing systems (SOAD). Because of the missing support for some basic SOA concepts in most M&S frameworks, difficulties arise when modeling and simulating service-oriented computing systems. Hence, Sarjoughian et al.⁵⁷ proposed an SOA-compliant DEVS (SOAD) simulation framework to address these issues. A DUNIP Web enables the DEVS framework as a service-oriented framework, but the M&S objectives are not necessary service-oriented systems. While a SOAD may not necessarily be service-oriented itself, the M&S objectives are service-oriented systems. The conceptual framework of an SOAD was reported by Sarjoughian et al.⁵⁷ Muqith et al.⁵⁸ extended the SOAD framework by introducing dynamic structure DEVS to model and simulate the structure changes in service-based systems.

From a service-orientation perspective, research on SOADs concerns the three roles of a SOA, messaging patterns, primitive and composite service composition, and hardware models for router links. *From an M&S viewpoint*, they compared and contrasted the SOA and DEVS. The DEVS framework is extended to support the concepts and capabilities of the SOA. The basic SOA roles, and the modeling of primitive and composite service composition, are investigated. Then, the hardware model of the network is introduced as a valuable complement to the software aspect of the SOA. Finally, an SOAD is implemented in a DEVJSJAVA environment and an example is illustrated to show the feasibility thereof. Ramaswamy⁵⁹ and Kim⁶⁰ model the roles and messages in an SOA with the classical DEVS formalism. Simulation experiments of a publish/subscribe SOA system are conducted to investigate the effectiveness. *Software/systems engineering* issues are not the focus of SOADs.

3.1.3. Web services based cell-DEVS framework (D-CD++). Wainer et al.²³ and Madhoun⁶¹ investigated a Web services based Cell-DEVS framework. The Cell-DEVS is a DEVS-based formalism that defines spatial models as cell spaces. Web enabling CD++, which is an M&S toolkit to execute Cell-DEVS models, can expose simulation functionalities as Web services to improve interoperability and reusability for the users' convenience. The architecture of a Web services based distributed simulation framework D-CD++ was shown by Wainer et al.²³ *From a service-orientation perspective*, the set of service interfaces in

D-CD++ includes session management, configuration, simulation modeling and control, and retrieving data interfaces. *From an M&S viewpoint*, the execution of D-CD++ conforms to parallel DEVS simulation protocols and adopts a global conservative time management strategy. The master and slave coordinators are used to reduce the number of exchanged messages among simulation services. Experiments and a performance analysis are carried out using D-CD++ over both the Internet and a dedicated fiber optic link. The results show that the overhead of SOAP messaging is the major bottleneck. In their latest work, Al-Zoubi and Wainer⁶² proposed an extension D-CD++ by using RESTful Web services middleware to perform distributed simulation.

3.1.4. Other related work and a summary. Other related work includes the non-hierarchical DEVSCluster-WS⁶³ based on Web services, and variable structure DEVS⁶⁴ as the basis for a dynamic SOA. The practice of SOA-based DEVS involves the testing of I/O behavior in services or systems⁶⁵ and network behavior analysis.^{66,67} Sun⁶⁸ improved the DEVS/SOA framework and investigated state management, time management, and a messaging scheme. However, the effectiveness, performance, and application of the framework need to be improved. Using the SOA concept and a new construct called the DEVS message namespace, Seo and Zeigler^{69,70} implemented an interoperable DEVS simulation environment in their latest research.

In summary, a formalism-based (e.g. DEVS) service-oriented simulation framework has the *advantages* of a rigorous theory basis and mathematical semantics. It is a general and flexible formalism framework that can model and simulate various systems. Many other formalisms and techniques (e.g. Petri net, state machine, UML, DoDAF) can be transformed or mapped into the DEVS formalism.^{22,71,72} The specification of DEVS and DEVSMML for models, DEVS simulation protocols with interface specification between simulators and models, the system entity structure, dynamic DEVS, and research on SOAs provide a solid foundation for service-oriented simulation. However, the DEVS framework has a possible *limitation* in that it is a too abstract and difficult formalism for users to follow. Difficulties also exist in interoperations with models and simulators for other formalisms. Although the DEVS standards organization is trying to standardize DEVS formalisms, model representation, model-solver interface, and model libraries,⁷³⁻⁷⁵ the standards have not yet matured, nor are they widely recognized and used by industry and academia. The primary focus of DEVS is on educational usage. Hence, the human computer interface and the

simplicity, convenience, and performance thereof need to be improved.

3.2. Model-driven framework

A framework of this type utilizes high level abstract models as the start and basis for the analysis, design, implementation, deployment, and maintenance in the entire lifecycle of service-oriented software development. The Dynamic Distributed Service-Oriented Simulation Framework (DDSOS)^{24,76,77} is a typical example. DDSOS is a distributed multi-agent service-oriented framework based on the Process Specification and Modeling Language for Services (PSML-S).⁷⁸ It has distinct functionalities such as dynamic simulation federation configuration management, automated simulation code generation, automated code deployment, multi-agent simulation for reconfiguration, and dynamic analysis. Furthermore, it is an M&S framework that supports rapid simulation, development, and evaluation of large scale systems. Jia and Zhang⁷⁹ proposed a similar framework. However, the differences between their framework and DDSOS are the replacement of PSML with UML as the common model specification and the lack of some dynamic properties.

From an M&S point of view, PSML-S is taken as the modeling language for SOA systems. The mappings from SOA and SOA workflows to PSML elements, structure models, and PSML models were reported by Tsai et al.⁴⁶ The mappings from HLA federation rules and interface specification to PSML were also investigated. RTI is taken as the runtime infrastructure. The optimistic time synchronization approach is used in the simulation engine whilst considering deadlock, synchronization, dynamic re-composition, and reliability. The *composability* of DDSOS is obtained by the Model Driven Architecture (MDA) method, while the *interoperability* is achieved by extending HLA/RTI.

From a service-orientation perspective, the services in DDSOS include system simulation agent services, environment simulation agent services, and RTI services. Once an application has been developed and deployed by DDSOS, three levels of reconfiguration are available, namely, rebinding, re-composition, and re-architecture. The dynamic properties of DDSOS are achieved by the core ideas of MDA. The simulation code can be automatically generated, deployed, and executed by modifying PSML-S models.

From a software/systems engineering viewpoint, the entire lifecycle is supported including modeling and specification, verification, code generation, validation, assembly and deployment, execution and monitoring, evaluation, and reconfiguration. The architecture and processes of DDSOS were reported by Fan.²⁴

DDSOS can completely support service-oriented systems engineering.¹² From an application perspective, to the best of our knowledge, DDSOS has only been applied to some preliminary cases.^{24,46,80}

The idea and *advantages* of model-driven, excellent dynamic composability and the full support of service-oriented systems engineering construct a solid foundation for service-oriented simulation. *Constraints* exist in that workflow-based behavior models of PSML⁸¹ are incapable of representing simulation systems that are not based on processes. In addition, PSML is not a widely recognized standard. DDSOS focuses more on the domain of service-oriented software development. It only extends some functionalities of RTI in simulation communities. Furthermore, it lacks some high level formalism or theory basis for PSML and the DDSOS framework. DDSOS provides dynamic properties. Meanwhile, it also causes difficulties with respect to efficiency, cost, and implementation. There is no support for mappings and automated transformation from other formalisms, e.g. from UML to PSML. The practice of DDSOS also needs to be extended.

3.3. Interoperability protocol based framework

This approach utilizes the some interoperability protocols (e.g. HLA)⁸²⁻⁸⁶ as the simulation bus for service integration and information exchange. A typical example is service-oriented HLA (SOHLA).¹⁵ SOHLA refers to the architecture enabled by an SOA and Web service (and other) techniques that support distributed inter-operating services. According to the layers of HLA, Web-enabled HLA can be implemented in four layers: at the communication layer (such as Web-enabled RTI^{87,88}), at the interface specification layer (e.g. HLA Evolved Web service API⁸⁹ and Unified Architecture⁹⁰), at the federate interface layer (such as the HLA Connector⁹⁰), and at the application layer (e.g. HLA Island⁸⁹). In the Swedish 'HLA and SOA integration' in support of the network-based defense, a prototypical architecture has been implemented and tested. This allows service-based and HLA-based systems to interoperate as shown in Figure 3.⁹⁰ This project integrates four federates using the native API, WS API, and HLA Connector respectively, which shows the feasibility of these approaches. Currently, the HLA Evolved Web service API is the latest development using SOA and Web service techniques to extend the HLA at the interface specification. The new generation of HLA standards named 'HLA Evolved'⁹¹⁻⁹⁴ has been published by the IEEE. Many leading commercial RTI corporations including Pitch and MAK are playing an active role in revising new HLA standards and developing or releasing new versions of RTI.⁹⁵⁻⁹⁷ Wang et al.¹⁵ surveyed the research and practice of a

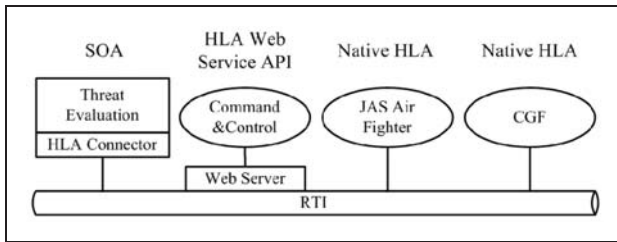


Figure 3. Architecture of Swedish 'HLA and SOA integration' in support for network-based defense.

SOHLA before the year of 2008. Latest work⁹⁸⁻¹⁰⁰ succeeded as well.

From an M&S perspective, the Base Object Model (BOM),^{101,102} modular Federation Object Model (FOM),^{94,103-105} and other enhancements improve the composability and flexibility of HLA simulation systems. From a service-orientation viewpoint, SOHLA reflects the idea of 'simulation as services'.¹⁰⁶ New improvements such as the HLA Evolved XML Schema,¹⁰⁷ smart update rate,¹⁰⁸ and fault tolerant mechanism¹⁰⁹ provide techniques to deal with problems of SOHLA in a net-centric environment. From a software/systems engineering perspective, the Federation Development and Execution Process (FEDEP),⁸⁵ and Distributed Simulation Engineering and Execution Process (DSEEP)⁴¹ need to be modified to reflect the Web centric idea¹¹⁰ and support for reuse, composition, and collaboration of services.

SOHLA has the *advantages* of a set of worldwide recognized IEEE standards, a broad audience, and the support of various products and applications by many vendors and organizations. Many future or legacy HLA-compliant simulation resources can easily be modified and reused in the new HLA standard. HLA has a solid research and practice foundation both in academia and the defense community. A recent peer survey¹¹¹ also reveals that the practical relevance and revision of HLA (e.g. HLA Evolved⁹¹) are still regarded as the future trends in distributed simulation. The *limitations* of SOHLA includes that HLA Evolved is the revision while not the revolution of HLA. The principles and semantics of HLA have not been exchanged. Some fundamental rules (e.g. monolithic FOM at the syntactic level) may constrain the further development of the HLA. In addition, the HLA only focuses on simulation interoperability and not on the composability of models or services. It also lacks a rigorous theory foundation. Conflicts also exist between coarse-grained services in an SOA and fine-grained services in HLA. Additionally, the HLA has the poor capability to support the composability and interoperability at semantic, pragmatic, dynamic, and conceptual levels.¹¹²

Some additional disadvantages and possible future directions were reported by Wang et al.¹⁵

3.4. EXtensible modeling and simulation framework

XMSF^{8,28} is defined as a composable set of standards, profiles, and recommended practices for Web-based M&S. XMSF utilizes Web services and related techniques to build up a common M&S technique framework. With the openness, dynamics, maturity, and scalability of Web service (amongst others) techniques, M&S can be integrated into operational systems in the GIG environment. Web/XML, Internet/Networking, and M&S are regarded as the major focus areas of XMSF. They cover the *M&S and service-orientation aspects* and have their requirements, focus, and related standards, respectively.^{8,113,114} SISO also created XMSF study group to deal with these issues.

The practice of XMSF includes the Web-enabled RTI¹¹⁵ and the project using XMSF to connect Navy Simulation Systems, Simkit, and CombatXXI, for joint modeling and analysis sponsored by SAIC.¹¹⁶ The Armed Forces of Korea also investigated the intelligent-XMSF approach based on autonomous Web services.¹¹⁷

The common technique framework for XMSF has the *advantage* of conceptual and technical support for service-oriented simulation. The related profiles of XMSF also provide experience in practice and implementation. The *limitation* of XMSF is its lack of concrete standards and implementation for service description, composition, and integration. It also lacks the support of software/systems engineering. In addition, the XMSF project has been terminated due to the lack of financial support for the XMSF study group since 2005. The product development group was not built up to develop the standards and products for XMSF.¹¹⁸ The focus of the members in XMSF changed to GIG and 'HLA Evolved' standards in net-centric environments.

3.5. Open grid services architecture based framework

The Grid is used to integrate various distributed resources as a 'Grid' in support of the sharing of collaborative resources and problem solutions for virtual organizations. Resource sharing is the essence of the Grid.¹¹⁹ Grids can be classified into computing, storage, data, knowledge, and service Grids according to the properties of the resources at the nodes.¹²⁰ There are two connections between an SOA and Grid computing. One is the application of SOA in Grid computing, where the Grid architecture such as the leading

OGSA is based on SOA. Another is the application of Grid computing to SOAs, that is, services are taken as Grid resources by Grid computing techniques to form a service Grid that supports sharing, management, and convenient access to services.

From a service-orientation perspective, Web services focus on the interface description and messaging of services, while Grid computing emphasizes the distributed computing resources including transparent access, fault tolerance, load balancing, and so on. The two techniques are complementary and are moving towards unification. OGSA is an architecture based on Web services and techniques. A Grid service is the extension of Web services that support stateful services. A new specification called the Web Services Resources Framework (WSRF)¹²¹ takes classical Web services as the interface to the stateful resources. *From an M&S point of view*, a framework called SOAr-DSGrid was proposed by the Nanyang Technological University Singapore for developing a component-based distributed simulation and executing the simulation in an SOA on the Grid.¹²² The Grid-based HLA management system (G-HLAM) by Rycerz et al.,^{123,124} the SOHLA RTI framework called SOHR by Pan et al.,¹²⁵ and the research of Xie et al.¹²⁶ implement HLA/RTI services as Grid services. Other computing and storage resources concerning M&S can be implemented as Grid services too. Zhang and Zhang¹²⁷ gave a detailed survey of a Grid-based distributed simulation. Zhang's dissertation¹²⁸ investigated the HLA RTI service, resource discovery service, simulation execution service, and simulation task migration issues in an OGSA environment. Li et al.²⁵ proposed a service-oriented GRID simulation called the Cosim-Grid.^{25,129} It is a service-oriented simulation framework based on HLA, Product Lifecycle Management (PLM), and Grid/Web services. Furthermore, it improves HLA on dynamical share, autonomy, fault tolerance, ability for collaboration, and security mechanisms. The prototype Cosim-Grid includes a resource layer, Grid resource service middleware layer, simulation application oriented middleware layer, an application portal layer of the simulation Grid, and application layer. Cosim-Grid extended the practice of OGSA based simulation framework. Li et al.²⁵ summarized the essence, architecture, key techniques, and practices of a simulation Grid in their latest review.¹³⁰

OGSA is a *valuable* complement to the state-of-the-art of service-oriented simulation frameworks from a resource management perspective. Research on Grid simulation provides the foundation for the reuse, distribution, and management of model components and other resources. The advantages also include the dynamic allocation and fault tolerance of resources, and also the transparency of computing resources to

the users. *However*, the M&S theory foundation, software/systems engineering, performance, and making full use of SOA in OGSA based service-oriented simulation frameworks need further research. Moreover, an OGSA needs some middleware such as Globus, while Web services are subjected to commercial standards and techniques. Additionally, the reliability, ease to use, and persuading numerous institutions to open their resources to the outsiders need to be improved.

3.6. Other service-oriented simulation frameworks

Besides the above classical frameworks, Northrop Grumman's Service Integration/Interoperation Infrastructure (Si3)^{131,132} was proposed to support simulation-based transformation. *From an M&S viewpoint*, composite simulation applications can be created through the integration and interoperation of models, simulation, applications, tools, utilities, and databases. *From a service-oriented perspective*, Si3 also provides a toolset to package applications as self-describing, discoverable, composable, and configurable services. Hence, it enables the integration and interoperation of independent, distributed heterogeneous applications. The conceptual and implementation architectures of Si3 were shown by Strelch et al.¹³¹ However, due to the limited literature known to the authors, the design and details of Si3 need further exploration.

Besides Si3, another service-oriented simulation framework for military purposes is the Web-enabled Joint Theater Level Simulation (JTLS).¹³³ It is used for conducting large-scale multinational exercises. Simulation operators (JTLS users) can participate in joint trainings using a Web browser to communicate with a JTLS game at a remote site anywhere in the world. *From M&S and service-oriented viewpoints*, the Web-enabled JTLS uses centralized model execution and computing styles while exposing configuration, message, and order management functions to Web services. It is a successful service-oriented war game simulation with a lower temporal and spatial resolution, lower update rate, and slower time advance rate.

In the enterprise application integration domain, international standards organizations and many researchers have investigated service description, publication and discovery, messaging and QoS based on Web services and semantic Web services.¹³⁴⁻¹³⁸ The related standards and publications have built the foundation for specification and supporting techniques. However, all these studies focused on a *service-oriented perspective*. Thus, the models, simulators, time management, and other issues in the M&S community need further research.

The research of Zhang²⁶ and Song²⁷ belong to the ontology or semantic driven service-oriented simulation framework. Ontologies or semantic Webs are used to improve the communication between users and Web services that use different terminologies.¹³⁹ These researchers proposed a conceptual framework based on an ontology or semantics, and investigated the issues of simulation service description, discovery, matchmaking, QoS-driven simulation services composition, dynamic simulations services composition, and fault-tolerance. In Zeigler and Hammond's new book,⁵⁶ an ontology and pragmatic framework was introduced to facilitate M&S based data engineering for net-centric environments. Yilmaz⁴³ presented an ontology-driven meta-level introspective Agent framework for improving dynamic composability. Lee and Kim¹⁴⁰ proposed a semantic Web based ontology framework that can reconfigure war game simulations on the fly by dynamically searching, discovering, and binding web services. Hu and Zhang¹⁴¹ presented an ontology-based collaborative simulation framework using HLA and Web services.

Ontology-driven approaches have the *advantages* of semantics-enriched service descriptions with possible capabilities of dynamic discovering, matchmaking, and binding. *However*, most research paid more attention to the service-oriented aspects (e.g. Web services or semantic Web services). The issues related to M&S, such as the verification, validation, and accreditation (VV&A) of simulation service composition, states management, and time management of the simulation service, presentation and implementation of M&S services, and the constitution and improvement of an ontology library for the M&S community, deserve further research.

4. Three-dimensional reference model: A unifying methodology for service-oriented simulation

Based on the comprehensive survey of various ad hoc service-oriented simulation frameworks, this section proposes a unifying framework or methodology (i.e. a three-dimensional reference model) derived from the review. It reveals the common functionalities and totality of research issues reflected in various specific frameworks.

4.1. Principle of the unifying methodology

The review on service-oriented simulations identifies (at least) three distinct, yet related fundamental dimensions (domains or viewpoints): M&S, service-oriented, and software/systems engineering. We regard the

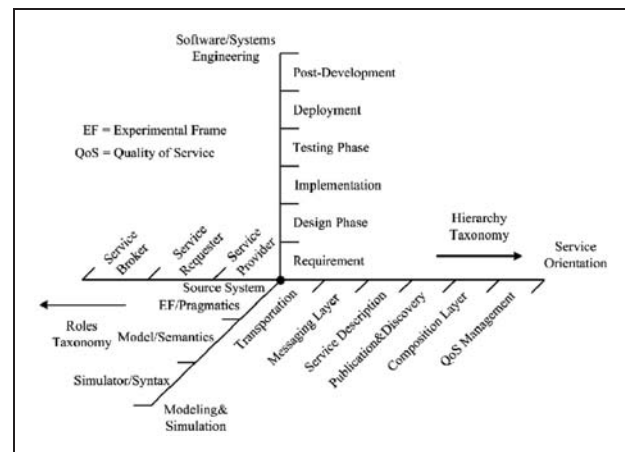


Figure 4. A reference model for service-oriented simulation.

three dimensions as independent or orthogonal domains/disciplines, since each has its own relatively complete and mature set of theory, approaches, standards, techniques, practices, and applications.

The three dimensions comprise a reference model for a service-oriented simulation (Figure 4). The M&S dimension is our focused basic domain. The SOA is a new paradigm/technology that impacts highly on the M&S, while the software/system engineering dimension can benefit the other two from a management view. Besides the dimensions, the elements in each dimension can also be derived from the review and are inspired by the fundamental concepts of each discipline. To reveal the common functionalities and totality of research issues reflected in various specific frameworks, we inspect the crossover of the three dimensions aggressively from a 1D, 2D, and 3D perspective. Our 3D reference model is also inspired by, but differs from, the methodology of Morphological Analysis¹⁴² and 3D morphology of systems engineering.¹⁴³ We pay more attention to the coverage of 'functionality morphology' in the 2D or 3D space, while not being constrained by the single cell focus of the Morphological Analysis method.

As stated before, there are two directions in the service-oriented simulation domain.¹⁸ One is the use of SOA in M&S, i.e. employing a service-oriented paradigm to extend the capacity of M&S techniques and frameworks. An example is the design and implementation of simulators that are services themselves, and can be invoked via SOA protocols.²⁰ The other is a vice versa approach, where M&S is used for SOA, i.e. M&S techniques are applied to address the problems in service-oriented systems. An example is the application of simulators that evaluate models of software packages designed along the SOA paradigm.^{76,77} Similarly, there are two levels in service-oriented simulation: the

problem to be simulated, and the simulation mechanism. The literature refers to the two levels as composability level (modeling, conceptualization, modeling question) and interoperability level (implementation, orchestration). Both levels can be service oriented. The reference model is intended to cover both directions and both levels by using different results that are Cartesian products from different orders of M&S and service-orientation dimensions.

Regarding the number and layout of dimensions, there may exist multi-dimensions, sub-dimensions or negative dimensions.¹⁴⁴ Service-oriented simulation must cover at least three dimensions such as M&S, service orientation, and engineering, as explained in the following subsections. In addition, other dimensions or sub-dimensions such as systems of systems¹⁴⁵ exist. However, it is hard to imagine and understand issues generated beyond three dimensions. Additionally, more than three dimensions may introduce great complexity and generate many meaningless cells in the crossover of every two or three dimensions. Thus, for simplicity, we do not include them here. Furthermore, any additional elements can be regarded as parts of the main three dimensions (e.g. systems of systems can be complementary to the engineering dimension). Moreover, the service-orientation dimension is broken down along a positive and negative axis. Hence, we stick to the three dimensions depicted in Figure 4.

4.2. One-dimensional implication

A 1D view enables us to look at each fundamental dimension individually. The source system is located at the origin. It stands for an existing or proposed system that we intend to observe or develop.

4.2.1. M&S dimension. Besides the source system, the basic entities in M&S³⁹ include an experimental frame (EF), model, and simulator. Modeling and simulation are the fundamental relationships. The EF–Model–Simulator comprises a general conceptual frame that explains nearly all the issues in the M&S domain well. The concepts implicated in the review also identify the three basic elements.

Additionally, other views in M&S, in particular composability and interoperability, are worth evaluating as well. They are complementary to Zeigler's EF–Model–Simulator frame. *'Composability is the capability to select and assemble simulation components in various combinations into valid simulation systems to satisfy specific user requirements'*.¹⁴⁶ Interoperability¹⁴⁷ is a precondition for composability; it is necessary, but not sufficient. Interoperability and composability are important criteria for M&S services to be communicated and

composed to meaningful systems. In particular, the work of Davis and Anderson,¹⁴⁸ the formal theory of semantic composability proposed by Petty and colleagues,^{146,149,150} the levels of conceptual interoperability model (LCIM) by Tolk and colleagues,^{32,151} the separation and contextualization of conceptual and simulation models by Yilmaz and Ören,^{152,153} and the work of others contribute greatly to this field.

The challenges and contributions on composability and interoperability lead to a hierarchical structure, in which we define three levels, i.e. Pragmatics–Semantics–Syntax. Pragmatics focuses on the use of information or artifacts within or across M&S solutions. Note that the EF is associated with pragmatics because it is the operational formulation of the M&S objectives. Semantics concentrates on the meaning of information or artifacts. It is the way in which we conceptualize our world as models. Syntax stresses formats and structures. It represents the way we implement and execute IT based simulation.

In addition, the Pragmatics–Semantics–Syntax hierarchy has a general sense. The syntactic and semantic composability,^{146,149,150} the LCIM,¹⁵¹ the layers of M&S,⁵⁶ the linguistic levels of dynamic system models,¹⁵⁴ and the interoperability challenges of model-based information systems (e.g. complex military simulation systems)¹⁵⁵ can also be mapped to these three levels with some reformulation or different interpretations. Although the same terms semantics and pragmatics convey multiple concepts, we highlight the meaning and use of information or artifacts within or across M&S solutions (depending on the spot where the information exchange happens) in this paper. Thus a simulation system can work meaningfully by itself or with other systems to serve a right purpose. Moreover, the three levels are also implicated in the reviewed approaches. The distributed computing infrastructural requirements (e.g. the SOA) for integration, distributed simulation requirements regarding the implemented systems for interoperability, and conceptual aspects and distributed modeling requirements for composability (composable M&S services) can also be associated with the three levels.

Consequently, the EF/Pragmatics–Model/Semantics–Simulator/Syntax comprise the M&S dimension from both an object-oriented view and the perspective of linguistic/conceptual information exchange. This gradually moves from conceptualization focused modeling views to implementation focused simulation views. With the complement of the Pragmatics–Semantics–Syntax, the M&S dimension is powerful enough to explain net- or Web-based alignment needs for distributed M&S services at different levels of composability and interoperability.

The major research activities in the M&S dimension include:

- capturing the conditions under which the system is observed or experimented;
- representing a physical, mathematical, or logical model of a system, entity, phenomenon, or process correctly;
- executing models correctly and efficiently;
- performing experimental design and scenario generation;
- collecting and validating the outcome of experiments;
- conceptualizing and representing reusable M&S components;
- evaluating and aligning the composability and interoperability of M&S components at different levels;
- publishing, searching, and selecting candidate M&S components;
- composing M&S components in various combinations into valid simulation systems; and
- validating the components and composite systems.

Related techniques are listed, such as:

- various modeling approaches;
- various simulation approaches and algorithms in the local, parallel or distributed execution paradigms;
- experimental design and scenario generation;
- data collection methods, and VV&A methods for models and simulators; and
- various standards and techniques for interoperability and composability, such as the HLA, BOM, MDA, and SOA.

4.2.2. Service-orientation dimension. Service-orientation is an increasingly state-of-the-art and promising approach for designing simulation systems. With the appealing characteristics of agility, reusability, and interoperability, services have been successfully incorporated in systems analysis, design, development, and integration.³ An implementation-independent service description can be published by a service provider via a service broker. Based on the published information, a service requestor can discover and compose requested services with other services. Service-oriented approaches can benefit business systems and others in addressing the requirements of agility and flexibility, while allowing for changes in the requirements themselves.¹⁸ The SOA³ is a conceptual framework for the design of business enterprise systems, while Web service¹⁵⁶ is the prevailing technology for implementing a SOA. Previous work³ provides a detailed review of approaches, technologies, and research issues in service-oriented approaches.

Service-orientation dimension has two taxonomies that come from the conceptual structure of SOA and the implementation hierarchies of Web services, respectively. The two taxonomies are complementary and the combination of them can better facilitate the analysis and implementation of service-oriented applications.

One of the taxonomies, from the viewpoint of roles, is structured as a triangle that consists of a service provider, requester, and broker. We use this particular order for this scale because the service provider and requester are more fundamental roles than the service broker. The service provider must provide its service earlier than the requestor's demand so as to compose a successful application.

The other taxonomy, from the perspective of Web service stack, is where the hierarchies of transportation, messaging, service description, service publication and discovery, composition and collaboration, and QoS management appear. Transportation, messaging, and service description are the core layers that constitute the basis for static SOA. Service publication and discovery, composition, and collaboration levels enhance the dynamic capabilities for dynamic SOA. QoS management makes services more dependable and robust by focusing on QoS requirements¹⁵⁷ such as performance, reliability, scalability, interoperability, and security. We sequence the elements by their decreasing importance on the scale in Figure 4.

4.2.3. Software/systems engineering dimension.

Simulation systems usually include software, at least in part.¹⁵⁸ The 'Simulation as software engineering' mode of simulation practice^{159,160} is applicable for teams of modelers and researchers, projects with lengthy lifecycles, and complex projects. For example, this model dominates military simulation due to the large scale models, long period of development, and expectation to be reused over a long period. The research and techniques for software engineering, especially software architecture and lifecycles, are of great use in simulation systems. The investigation of McKenzie et al.¹⁵⁸ showed that there are no fundamental differences at the architectural level between simulation systems and general software systems. Formal and informal software architecture design methods can also be widely used in the M&S community.

Additionally, systems engineering can also benefit service-oriented simulation as a valuable complement¹⁶¹ in the hardware, optimization, trade-off, decision making, and other aspects that fall beyond the scope of software engineering. Systems engineering is a multidisciplinary methodology that comprises several logical phases that are independent of ad hoc techniques. In general, the phases define that each system goes through a lifecycle, and certain steps need to be followed to ensure that the

objective is supported. The better our system is managed in the phases, the smoother it runs.

The lifecycle of software/systems engineering may be assigned to different ontologies from multiple viewpoints.^{38,162} In this work, we use the taxonomy of requirement, design (e.g. description, design, and analysis), implementation, testing, deployment, and post-development (e.g. maintenance, evolution, reuse, and retirement). In fact, the activities included in the engineering dimension are often cyclic or concurrent.

The research and practice of software/systems engineering were reported by Jamshidi,¹⁴⁵ Mei and Shen,¹⁶³ and Hitchings.¹⁶⁴ Note that design and implementation often receive preferential treatment in general research and practice.

4.3. Two-dimensional implication

While the 1D approach considers each dimension individually, a 2D view inspects domains consisting of the Cartesian product of two dimensions to reveal the systematic cross-discipline landscape of service-oriented simulation. It also reveals the gaps in the current research of service-oriented simulation systems.

For a given specific framework that is compatible with the reference model, the issues resulting from the reference model can be categorized as the following three categories:

1. *core* issues (C), the fundamental nature of service-oriented simulation; if they are not present, the framework cannot be called a service-oriented simulation framework;
2. *supporting* issues (S), the important characteristics of service-oriented simulation; if they are missing, the framework will be heavily affected; and

3. *nice-to-have* issues (N), the complementary functions of service-oriented simulation; if they are not present, the framework may be slightly affected.

This classification can be applied to 1D, 2D, and 3D views. The crossover between research disciplines is identified and analyzed in Tables 1, 2, and 3. The 2D tables can be used for a cross-consistency assessment¹⁴² process. They identify the logical and empirical meaning of each cell that consists of a pair of elements from the compared dimensions.

4.3.1. Narrow service-oriented simulation. The Cartesian product of the M&S and service-orientation dimensions allows us to treat service-oriented simulation in a narrow sense (Table 1). This is the fundamental domain for service-oriented simulation, which we refer to as the ‘narrow approach’ since it may lack rigorous engineering principles or processes. Some ad hoc research or practices^{79,131,132} belong to this category. This 2D space has two implications that reveal the two directions of SOAs for M&S and vice versa: an approach that enables the extension of traditional M&S artifacts by service-oriented principles, and an approach that models or simulates service-oriented systems by means of M&S. For example, on the one hand, we can use SOA artifacts to publish a model as a service; on the other hand, we can also model SOA artifacts for analytical purpose. As mentioned previously, the Cartesian product of differently sequenced dimensions provides different directions. This principle is an extension of the non-directional cross-consistency assessment process.¹⁴²

Capturing M&S and SOAs as discrete phases and crossing them, produces some interesting observations. (i) From a 1D view, the headings of the first column in

Table 1. Narrow service-oriented simulation (M&S vs Services)

	Broker	Requester	Provider	Transport	Messaging	Description	Publish and Discovery	Composition	QoS
EF/Pragmatics	N	N	N	N	N	N	N	N	N
Model/Semantics	S	C	C	C	C	C	S	S	N
Simulator/Syntax	S	C	C	C	C	C	S	S	N

The increasing gray intensity of the cells identifies nice-to-have (N), supporting (S), and core issues (C), respectively. EF = Experimental Frame.

Table 2. M&S engineering (M&S vs Engineering)

	Requirements	Design	Implementation	Testing	Deployment	Post-development
EF/Pragmatics	N	N	N	N	N	N
Model/Semantics	S	C	C	S	S	N
Simulator/Syntax	S	C	C	S	S	N

The increasing gray intensity of the cells identifies nice-to-have (N), supporting (S), and core issues (C), respectively. EF = Experimental Frame.

Table 3. Service-oriented engineering (Services vs Engineering)

	Broker	Requester	Provider	Transport	Messaging	Description	Publish and Discovery	Composition	QoS
Requirements	S	S	S	S	S	S	S	S	N
Design	S	C	C	C	C	C	S	S	N
Implementation	S	C	C	C	C	C	S	S	N
Testing	S	S	S	S	S	S	S	S	N
Deployment	S	S	S	S	S	S	S	S	N
Post-development	S	S	S	S	S	S	S	S	N

The increasing gray intensity of the cells identifies nice-to-have (N), supporting (S), and core issues (C), respectively.

Table 1 identify the discrete elements together with their relationships in the M&S dimension. This principle also works in the SOA dimension. From a 2D view, a cell in the 2D table reflects a sequential pair of elements from the crossover of the two dimensions. For example, the simulator is where the simulation relation is captured. The simulator can certainly be a service with all core SOA capabilities. (ii) Furthermore, from the composability view of Pragmatics–Semantics–Syntax, if the assumptions and constraints regarding service description differ, we will not be able to discover the services. If we use different semantics to describe the services, we cannot compose them to work correctly. (iii) Moreover, the M&S dimension can be further discretized as a conceptual model, simulation model, and context.^{152,153} A SOA also has other detailed taxonomies. The crossover of further discrete elements with their new relations can facilitate deeper research on the reusability and composability of M&S services.

4.3.2. M&S engineering. The Cartesian product of the M&S and software/systems dimensions provides an M&S engineering domain (Table 2) that applies engineering principles to traditional M&S as in, e.g., the classical HLA FEDEP⁸⁵ and VV&A standards.⁸⁶ This is the traditional M&S engineering domain that does not necessarily refer to service-oriented simulation.

On the one hand, M&S engineering demands all elements of EF/Pragmatics–Model/Semantics–Simulator/Syntax to be addressed in each phase of the software/system engineering. For instance, testing in net-centric environments needs to be conducted simultaneously at the pragmatic, semantic, and syntactic levels.⁵⁶

On the other hand, M&S engineering also demands each element of EF/Pragmatics–Model/Semantics–Simulator/Syntax to be supported and aligned in and between all phases of the engineering process. For example, conceptual views in the requirements phase will influence the reuse of the system in the post-development phase. This allows requirements (e.g. composability and interoperability) that come up in later phases to be formulated and supported in

earlier phases. Otherwise, we are disconnected if our metrics for success when we define the system are different from the metrics for success when we test the prototype and later the real system. Note that the EF/pragmatics changes over the phases of the systems engineering process. It first specifies the objectives, assumptions, and constraints for requirements, becomes a development context later, then turns into a reference for testing cases, and finally becomes the context for VV&A and post-development.

4.3.3. Service-oriented engineering. The Cartesian product of service-orientation and software/systems dimensions creates a service-oriented engineering domain (Table 3). Here, engineering principles are applied to a service-orientation community. Although the basic engineering principles seem still unchanged (along the classical engineering dimension), new requirements and challenges are introduced by the SOA paradigm. For example, services are key elements, service interfaces, reuse and composition are paid more attention to, and the development style is mainly model driven. Service-oriented engineering is a new emerging domain. Typical examples include service-oriented systems engineering¹² and service-oriented software engineering.^{13,14} In particular, these authors discussed the impact of the SOA paradigm on classical software/systems engineering principles and practices.

4.4. Three-dimensional implication

Despite the partial perspective for the 1D and 2D interpretation, the 3D view illustrated in Figure 5 provides a complete multi-perspective consideration of a service-oriented simulation. The whole 3D space consists of the Cartesian product of all three dimensions. The 3D space can be illustrated as a cube, with each cell representing part of our knowledge. The importance of each cell is identified according to the core, supporting, and nice-to-have classification. The coverage of cells indicates our active areas of the totality of research issues in service-oriented simulation. This cube represents ‘service-oriented M&S engineering’, also called ‘general

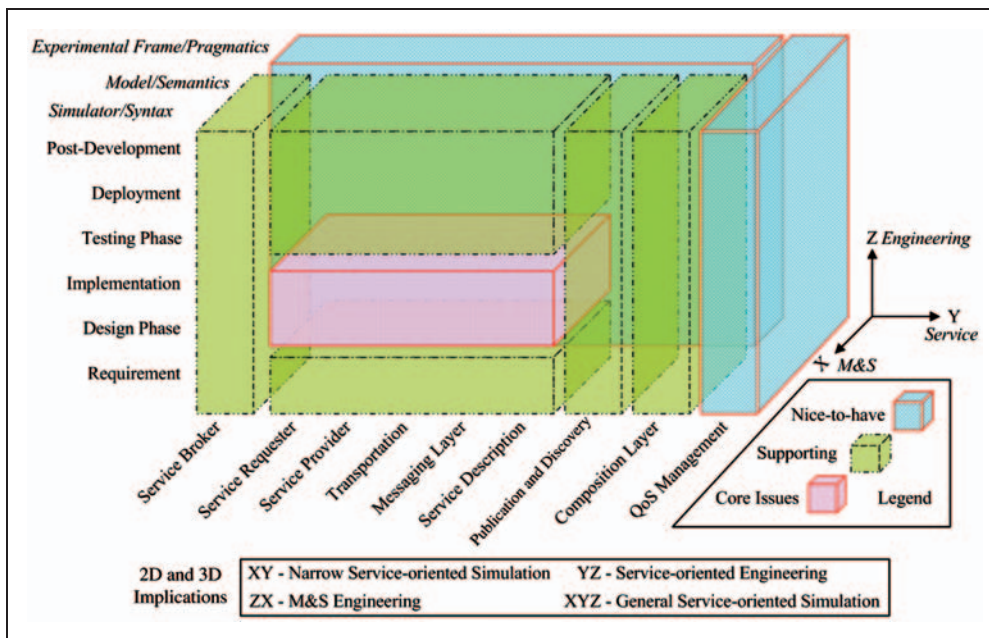


Figure 5. Three-dimensional reference model for service-oriented simulation.

service-oriented simulation' because it applies engineering principles to the whole development lifecycle of service-oriented simulation systems. The cube identifies several axes for necessary alignment, and is able to show and explain nearly all the challenges in service-oriented simulation, within and across phases (software/system engineering), solutions (services), and concepts (M&S). The 3D model can facilitate communications in and across organizational or disciplinary boundaries, in particular among managers in engineering, implementers of solutions, and specialists in M&S. In summary, to evolve as a new and mature M&S paradigm, service-oriented simulation must cover the whole 3D space demanded by the 3D model.

The 3D reference model can be applied to separate concerns and used as a taxonomy to find the similarities and differences of the existing service-oriented simulation frameworks. Moreover, it can aid domain experts to define clearer and more specific activities. Some sub-phases or steps can be added by multiple discipline experts using Cartesian products so that potential new research issues can be discovered. Examples of possible research problems generated by the crossover of the service-orientation and M&S dimensions include how to encapsulate the capability of models, simulators, and experimental frames as services, and how to manage, use, and implement them in their respective layers. From an engineering point of view, the properties, design, and implementation problems should be considered as complements to the above issues.

4.5. Descriptive and prescriptive roles

Engineering methods distinguish characterization (description) and mandatory (prescription).¹⁶⁵ The 3D reference model for service-oriented simulation can serve both functions. In its descriptive role, the 3D model describes the properties and functional morphology of service-oriented simulation within an existing ad hoc framework. In its prescriptive role, the 3D model prescribes a set of net-centric M&S requirements that must be satisfied during the engineering of a proposed specific framework.

The descriptive role of the 3D model can be used to depict or analyze the abilities and maturity supported by the existing service-oriented simulation frameworks (such as those surveyed in Section 3). It can also show the coverage, similarities, and differences of issues addressed in the 3D space by ad hoc frameworks. We placed all the surveyed approaches in the 3D space and conducted a series of detailed comparisons from 1D, 2D, and 3D views as explained in Section 5. This indicates that the 3D model can describe various service-oriented simulation frameworks. In contrast, the solid coverage properties of all the reviewed approaches justify the 3D unifying framework.

The prescriptive role can be utilized to prescribe a set of issues or requirements that must be satisfied in order to develop a proposed service-oriented simulation framework. Researchers can analyze, select, and synthesize specific issues, formalisms, and approaches based on their requirements. In particular, it reveals

valuable gaps or strategic migration paths for each approach to increase the maturity of service-oriented simulation. Using the prescriptive role, we present some recommendations in Sections 5 and 6.

The two roles can be combined to show the potential and possible future directions of classical service-oriented simulation frameworks. The first role shows which cells have been covered in the 3D space, while the other shows which cells need to be filled in. The 3D model emphasizes applying rigorous engineering principles and methods to embrace the full potential of service-oriented simulation.

5. Applying the unifying methodology to describe, compare, and prescribe the reviewed frameworks

Based on the review and the 3D reference model, this section shows the functional benefits and recommended practice of the unifying methodology to describe, compare, and prescribe various frameworks. It reveals the 3D reference model as a guideline to find the contributions and gaps of the reviewed approaches. It also indicates the application of the 3D model for frameworks selection and gaps filling.

5.1. Recommended practice of the unifying methodology

The unifying framework can be built from existing approaches by merging the similar, equivalent, or complementary capabilities they provided in all three dimensions. Therefore the unifying methodology has the functional benefits to (i) identify the contributions and gaps of existing approaches, (ii) facilitate the selection of frameworks, (iii) show how the gaps may be closed by expanding or merging of existing part solutions, (iv) check the alignment of system activities between all dimensions, or (v) establish new research topics where the gaps are not closed by any contribution. The recommended practice of the unifying methodology is listed as follows.

1. *Define objectives of a service-oriented simulation.* The purpose of this step is to identify user needs and develop objectives. The direction of the service-oriented simulation should be determined based on the problem to be simulated and the proposed simulation mechanism. If the problem is service oriented but the mechanism is not, then it belongs to M&S for SOAs. Vice versa, if the mechanism is service oriented but the problem is not, then it belongs to SOAs for M&S. If both are service oriented, then service-oriented approaches are used to

simulate service-oriented applications. If neither is service oriented, it belongs to classical M&S that is beyond the scope of this paper.

2. *Develop capability requirements.* Based on a conceptual analysis of the problem, this step is intended to identify all the required capabilities in the 3D space.
3. *Select candidate frameworks for reuse.* The purpose of this step is to determine if an existing reusable framework meets or partially satisfies the requirements. The descriptive and prescriptive roles of the 3D model can be used to identify the capabilities provided and gaps left by the current frameworks. Sections 5.2, 5.3, and 5.4.1 describe this activity in detail.
4. *Expand or compose known capabilities of candidate frameworks.* This step is intended to close the gaps by merging the existing part solutions of all necessary dimensions. We give a detailed explanation in Section 5.4.2.
5. *Align system activities between all dimensions.* The purpose of this step is to dissolve the conflicts of system activities between all dimensions when a best framework is reused or some candidate capabilities are composed. Section 5.4.2 presents this activity in detail.
6. *Establish new research topics or develop new capabilities.* This step is intended to identify and suggest research and practice topics for the missing gaps that have not been covered by any contribution. We give a detailed explanation in Section 5.4.2.

The functional benefits of the unifying methodology are revealed in the lifecycle of the recommended practice above. It facilitates to identify what do we need (required capabilities in step 1 and 2), What do we have (capabilities provided by the existing frameworks in step 3), What do we miss (gaps in the frameworks, plus missing alignments if the frameworks address different dimensions in step 3), and how do we close the gaps (expanding or merging the capabilities of frameworks in step 4, alignments of system activities between all dimensions in step 5, and identifying remaining gaps for future research in step 6).

The recommended practice can be tailored to meet specific user needs. We apply the steps to the reviewed frameworks in the following subsections. Note that we focus only on the techniques aspect that covers the both directions and the whole 3D space demanded by service-oriented simulations. Therefore we would not mention step 1 and 2 in the following subsections that can be found in tailored domain-specific problems of the ad hoc frameworks.^{22,24}

5.2. Contributions of existing frameworks

The 3D reference model can be utilized as metrics to find and compare the capabilities provided by the existing frameworks. We fill the 3D space with existing blocks of capabilities and make a detailed analysis from all the 1D, 2D, and 3D views in the Appendix Tables A1–A9. A preliminary usage of the 3D model to the DUNIP and the DDSOS frameworks has been reported in our previous work.¹⁶⁶ Additionally, we give an overall comparison of the reviewed frameworks in Table 4. We compare the six reviewed categories of approaches, listed as rows, with respect to the metrics for typical examples, three important dimensions, as well as pros and cons. In particular, we specifically check the model specification, M&S standards, and simulation protocols in the M&S dimension; resources that are published as services and interfaces, dynamic composition, fault-tolerance, QoS management, and semantic UDDI of services in the SOA dimension; and lifecycle support in the engineering dimension. Using the descriptive role of the 3D model, we can identify the contributions of existing frameworks from each ad hoc framework in particular and the union of all the frameworks in general.

From the viewpoint of each ad hoc framework, the formalism-based approach has a rigorous theoretical basis and a number of important properties such as modular and hierarchy composition. This approach has an extensive coverage (especially in the M&S dimension) in the 3D space. With similar wide coverage, the model-driven method pays more attention to the direction of ‘M&S for SOAs’ (e.g. service-oriented software engineering and dynamic properties). Although the coverage seems inadequate, the interoperability protocol based approach has a mature basis of international standards, wide applications, and promising potential. In spite of weak coverage, the XMSF is the earliest approach amongst others that outlines the techniques framework for Web-based simulation. The OGSA-based method supports dynamic management, reuse, and transparent access for various M&S resources. The coverage of this method is moderate and needs further investigation. The Si3 and ontology-driven frameworks have the advantages of service integration and semantic interoperability respectively. Their coverage indicates the emphasis on the publication, discovery, and composition of services.

From the perspective of the union of all the frameworks, the existing capability blocks (Appendix Tables A1–A9) are intensively distributed and overlapped in the core area. The supporting space has a moderate coverage, and the nice-to-have field is distributed by some sparse capability units. The united capabilities of

all the frameworks lead to a better coverage of the whole 3D space. This indicates the existing frameworks are developing aggressively from core, to supporting, and nice-to-have regions. In the future, the frameworks or the union of them are expected to provide full capabilities that can fill in the 3D space completely. Meanwhile, different frameworks have different concerns. The unique or scarce capabilities they provided indicate their competitive edge. Note that there is a sharp distinction between the directions of ‘SOAs for M&S’ and vice versa. Different directions or objectives may make the semantics of the cells different, and bring difficulties to the merging and alignment activities. Although our 3D model can cover both directions, we pay more attention to the use of SOAs for M&S.

5.3. Gaps of existing frameworks

Using the prescriptive role of the 3D model as well as the text in the review, we can identify the gaps of existing frameworks. The formalism-based approach is limited in terms of standardization and ease of use. The gaps in the 3D space show room for improvements such as the publication and discovery, the composition, the broker, the QoS, testing, and post-development. The model-driven method has limited capability in terms of M&S. Its gaps represent that the requirements, SOAs for M&S, conceptual interoperability aspects amongst others can be further improved. The wide gaps of interoperability protocol based approach demand the enhancement of model services, higher levels of composability, and full potential of SOAs. The XMSF needs concrete standards and implementations, and the XMSF study group has been dismissed. The OGSA-based method requires a Grid middleware infrastructure. The gaps indicate the M&S aspects and full potential of SOAs require further research. Regarding the Si3 and ontology-driven frameworks, the gaps show that the M&S domain, the VV&A of services, and full lifecycle support need to be improved.

5.4. Recommendations for frameworks selection and gaps filling

5.4.1. Frameworks selection. Based on the capability requirements, and the contributions and gaps of existing frameworks, some recommendations can be made for the selection of frameworks. Technical constraints (e.g. reusability, VV&A, standardization) and managerial constraints (e.g. security, availability, preference, and mandate) should be considered before the selection process. In general, the framework which meets the requirements with maximum capabilities under all the constraints is the best choice. Otherwise, a set of

Table 4. Overall comparison of classical service-oriented simulation methods

Methods	Examples	M&S	Service-orientation	System Engineering	Advantages	Limitations
Formalism based	DUNIP, DEVS/ SOA, SOAD, D-CD++	Unified DEVS model specification. DEVSML for platform independent models. SOAD can model and simulate service-based software and hardware systems. DEVS simulation protocol.	Simulators as services, models as resources in DUNIP. No coordinator services. Session management, configuration, simulation modeling and control, and retrieving data service interfaces in D-CD++.	DUNIP has bifurcated model-continuity systems engineering methodology.	Mature formalism with long history. Strong presentation capability to various systems. Rigorous theoretical basis and mathematical semantics.	Too abstract and hard to follow by users. Have not been widely recognized by industrial and academic standards. Primarily for educational use. Simplicity, convenience, and performance need to be improved.
Model driven	DDSOS	PSML-S can model SOA systems. RTI as runtime infrastructure. Optimistic time synchronization.	Systems/environment simulation agent services and RTI services. Support dynamic rebinding, re-composition, and re-architecture.	MDA and service-oriented systems engineering (SOSE) support.	Model-driven, excellent dynamic composability, and SOSE support.	Focus on service oriented software development. Limited simulation capabilities. Theory, efficiency, and applications to be improved.
Interoperability protocol based	Service oriented HLA, HLA Evolved Web Service API, etc.	BOM and modular FOM facilitate interoperability levels of models. Low bandwidth, uncertainty, and dynamic properties need considering.	Web-Enabling HLA for communication, HLA interface specification, federate interface, and application layers. HLA Evolved XML Schema, smart update rate, and fault tolerant mechanisms.	FEDEP needs to be modified to reflect the idea of Web centric and support of reuse, composition, and collaboration of services.	Worldwide recognized IEEE standards. Solid research and practice foundations. HLA Evolved new standards	Revision while not the resolution of HLA may constrain further development. Lower levels of interoperability. Conflicts between SOA and HLA in service granularity.
XMSF	XMSF and profiles	The M&S focus area of XMSF.	The Web/XML, Internet/ Networking focus area of XMSF.	N/A	Pioneer technical frame; separate focus areas, issues, and techniques.	Lack of concrete standards, products, and systems engineering support. Has been terminated.
OGSA based	Cosim-Grid, SOA-DSGrid, G-HLAM, SOHR	Simulation components, HLA/RTI services, computing, and storage resources can be Grid services.	Focus on management of distributed computing resources. Based on Grid middleware.	Not clear	Resource dynamic allocation, load balancing, and fault tolerance. Transparency.	Needs grid middleware. M&S theoretical basis, systems engineering, performance, and full use of SOA to be improved.
Other approaches	Si3, ontology/ semantic driven framework	HLA/RTI simulation engine in Si3. Service description, semantic service matchmaking, not focusing on simulation execution in ontology approach.	Si3 packaging models, simulation, applications, tools, utilities, and databases as services. Ontology method focuses on service UDDI, composition, and fault-tolerant.	Have some development and usage procedures.	Integration and interoperation of heterogeneous applications in Si3. UDDI and semantic composition in ontology methods.	Few publications and not mature. Need further investigation in the M&S dimension, especially VV&A, states, and time management of simulation service.

frameworks that partially satisfies the requirements can be considered as candidates for composition and alignment.

In particular if the theory or education purposes are in a dominate position, the formalism-based approach should be considered the first. If the problem and objects are SOA systems, and the dynamic properties of service-oriented software engineering are emphasized, then we can choose the model-driven framework. If the governments or managers mandate mature standards for interoperation and compatibility with legacy systems, the interoperability protocol based approach like HLA is an appropriate choice. If we highlight the sharing of resources and problem solutions for virtual organizations, we can give the first priority to the OGSA-based approach. We would not recommend the XMSF approach because it has been ceased. The ontology driven framework can be considered if conceptualization of domains, semantics of services, brokers, publication and discovery are emphasized. Please note that the ontology and Si3 frameworks are not as mature as the others thus far.

According to Tables 4 and A1–A9, and also the viewpoint of using the SOA paradigm to extend the capability of M&S frameworks, we found that the formalism-based approach is the most mature one from an M&S theory perspective. On the other hand, the interoperability protocol based method has the most potential in the practice. Combining the two approaches with rigorous engineering methods^{44,165} can better facilitate the meaningful composition and interoperation of simulation services, and promote the theory and practice of service-oriented M&S.

5.4.2. Gaps filling. After the selection process of candidate frameworks, this subsection discusses the ways to fill the remaining gaps by the possible expansion/composition, alignments, and recommendations for future research.

1. *Expanding or merging of candidate part solutions.* There are two possible ways to fill the missing gaps. One is the extension of the best candidate framework itself. For example, the SOHLA framework could be extended to fill the missing gaps by providing object models as services. The other is the merging of part solutions provided by a set of candidate frameworks. Some application-independent capabilities of candidate frameworks can be reused as common services, such as the runtime infrastructure services from the SOHLA, and the broker service from the model driven framework. The merging of the two frameworks can benefit

the SOHLA from the publication and discovery of its service description. The reference model can facilitate to identify the expansion or merging path in the 3D space. In this step, the merging of model services and disposing of duplicated/similar services are difficult problems that need further research.

2. *Alignment of system activities between all dimensions.* In course of expansion or merging candidate frameworks, the alignments between system activities take place. This step is intended to check and align the compatibility and consistency between capability units or system activities in terms of objectives, assumption, and constrains. It is more important when heterogeneous capability blocks are composed. The 3D reference model acts as a checklist for alignment along each column and row in the 1D, 2D, and 3D Tables A1–A9. From the 1D view, the composition and artifacts of capability units are aligned along the M&S, service-orientation, and engineering dimensions for compatibility and consistency. From the 2D view, system activities between all dimensions are checked. For example, in Table A5 the EF/Pragmatics are aligned from the requirements to post-development phases; and the pragmatics, semantics, and syntactic are adjusted simultaneously in the testing phase. From the 3D view, all the capability units are adjusted and harmonized across phases, solutions, and concepts. The alignments at the syntactic, semantics, and pragmatics levels by using the data, process, and assumption-constraint engineering were reported by Tolk et al.⁴⁴
3. *Remaining gaps for future research.* There are still some gaps that have not been covered by any contribution. This precludes the union of existing frameworks from a full coverage of the 3D space. In the M&S dimension, the gaps indicate that the capability of ‘models as services’ falls some way short. In the service-orientation dimension, the gaps of brokers, publication and discovery, dynamic properties, composition, and QoS still have room for improvements. In the engineering dimension, the full lifecycle support can also be further enhanced, in particular the phases of requirement (e.g. the semi-automatic generation of models, EFs, or generation of testing frames from requirements), testing, deployment, and post-development. The gaps also reveal that the higher levels of conceptual interoperability are inadequate. Therefore the M&S services are not so well annotated to facilitate intelligent agents to find, understand, orchestrate, and compose services meaningfully and automatically.

6. Conclusions and recommendations

6.1. Conclusions

As the requirements for M&S interoperability, reusability, and composability in net-centric environments are continually being extended, simulation systems are becoming more standardized, and are favoring the introduction of components, hierarchies, networks, and services abilities.¹⁶⁷

Service-oriented approaches offer functional benefits for improving the way information resources and functional capabilities are shared, collaborated, and integrated.¹⁶ Service-oriented M&S is the interdisciplinary field of M&S, the service-oriented paradigm, and software/systems engineering. It addresses the interoperability and composability challenges of distributed M&S services and represents the current focus and future direction of M&S in the prevailing net-centric environments. Service-oriented M&S can extend the capabilities of classical frameworks by addressing the accessibility, reusability, composability, extensibility, fault-tolerance, intellectual property, deployment, maintenance, and agility of heterogeneous M&S resources and those from other domains.^{8,22,24} Service-oriented M&S is also an important approach for dealing with ambiguity, uncertainty, and variability¹⁶⁸ of complex systems in a net-centric environment. Service-oriented M&S has important research value and great application potential.

In this paper, we presented a comprehensive review of various service-oriented simulation approaches. We also proposed a three-dimensional unifying framework or methodology derived from the review. On the one hand, the review shows how M&S and SOA are combined in different ways to address different issues, and also the merits and limitations of each approach. The review and comparisons in Section 5 are useful for assisting researchers in selecting the appropriate method for their specific purpose. Furthermore, the review facilitates the classification, evaluation, selection, implementation, extension, and application of the reviewed or future frameworks. On the other hand, the 3D unifying framework is a general (or high level) systematic methodology for service-oriented simulation derived from the current state-of-the-art. It shows the intersections of three basic disciplines (i.e. M&S, SOA, and software/systems engineering) to address interoperability and composability challenges of distributed M&S services. It reflects the common functionalities and totality of research issues coming from various ad hoc approaches. It also facilitates describing, analyzing, developing, and addressing issues in service-oriented simulation in a systematic way, and can be used as a guideline or an analytic means for finding potential and possible future directions for

various specific frameworks. Proposed new frameworks and the extension of current frameworks can also benefit from our 3D methodology and comprehensive survey.

6.2. Recommendations

Although research on service-oriented simulation frameworks has made great progress, there are still many unsolved problems and challenges. We list some of these and provide recommendations based on the proposed 3D methodology for future directions.

1. *The body of knowledge of service-oriented simulation.* To allow service-oriented simulation to succeed object-oriented simulation³⁵ in the general sense, the body of knowledge,¹⁴⁴ rigorous theoretical basis, and related standards, techniques, and implementations need to be built up. The intension and extension of the concepts need to be investigated. Moreover, although the formalism-based and model-driven approaches can cover both the directions of M&S for SOA and vice versa, the body of knowledge may not be complete until the gap in the first direction (M&S for SOA)¹⁶⁹ is also comprehensively investigated. The service modeling approaches, VV&A, and experiments, amongst others, should also be addressed.
2. *The three-dimensional reference model.* The 3D reference model is a unifying framework and methodology for service-oriented simulation derived from the reviewed approaches. It reflects the common functionalities and totality of research issues of service-oriented simulation through a crossover of the disciplines of M&S, service-orientation, and software/systems engineering. It identifies the core, supporting, and nice-to-have services. The 3D framework has both description and prescription uses. The evaluated research and results thus far already justify the framework. We highly recommend that researchers use the 3D methodology for describing, analyzing, developing, and addressing issues in service-oriented simulation in a systematic way. We also recommend the use of the two roles of the 3D methodology to find the potential and possible future directions of various ad hoc frameworks and the similarities and differences thereof. Nevertheless, we would like to provide possible challenges to support an extensive further study of the 3D framework. (i) *Regarding techniques for the 3D model.* Detailed evaluation criteria can be given for each cell in the 3D space. A formal (e.g. mathematical) presentation and proof of the 3D model and its evaluation criteria are also welcome.

- The maturity degree of coverage for each cell (e.g. full, partial, none) can be further investigated. Consistency of the multiple views and the qualitative and quantitative research of the 3D frame are also interesting issues. (ii) *Regarding the use of the 3D model.* Domain experts can define, modify or add more clearly, specific activities in the 3D model. Interdisciplinary experts can utilize the model to discover further new research issues through the Cartesian products of three or any two dimensions. (iii) *Regarding the philosophy of the 3D model.* The SOA is a paradigm and technique that can access and execute distributed computational capabilities. It can also bridge various techniques (e.g. the DEVS, MDA, HLA, Grid, XMSF, and so on). We focus on the SOA paradigm for M&S and the way SOAs and M&S are combined in this paper. However, when the SOA is a thing of the past, the philosophy of the 3D research can still be applied directly or with slight modification by replacing the SOA dimension with another emerging paradigm.
3. *M&S dimension.* From an M&S point of view, most research that the authors are aware of exposes the functionalities of the simulation infrastructure as services, which reflects the idea of ‘simulation as services’.¹⁰⁶ However, from a modelers’ perspective, exposing simulation models as services is also necessary and important for intellectual property and model reuse. The DEVS modeling specification, DEVSMML, and BOM standard provide solid foundations for the standardization of model services. However, conflicts between reuse granularity¹⁷⁰ and execution performance need to be solved. Further interesting research issues are the standardization of simulation services, VV&A of stand-alone services and composition services, performance,¹⁵ multicast ability over wide area networks,¹⁷¹ semantically lossless interoperation and composition of services,¹⁴⁸ and the interoperation with Commercial-Off-The-Shelf simulation packages.¹⁷²
 4. *Service-orientation dimension.* From the perspective of service-orientation, to make full use of the SOA², supporting simulation components for registering, finding and dynamic integration with UDDI, enhancing the service orchestration and choreography, and introducing properties of a dynamic SOA are worth further investigation. In addition, the QoS, fault-tolerance, and automated or semi-automated service interoperation and composition¹⁷³ deserve further investigation.
 5. *Software/systems engineering dimension.* Service-oriented simulation systems should utilize rigorous engineering principles to facilitate the reuse and composition of complex applications. Standard

software/systems engineering approaches^{38,162} should be investigated to guide the analysis, design, implementation, and application of service-oriented simulation frameworks. Engineering principles in the M&S domain, e.g. FEDEP⁸⁵ and DSEEP⁴¹ with extensions to the service-oriented paradigm, can also facilitate the research of service-oriented simulation engineering.

6. *Research and practice of service-oriented simulation frameworks.* The classical service-oriented simulation frameworks surveyed in this paper all have their pros and cons. We recommend that researchers select, compose, align, or extend appropriate candidate frameworks for their specific purpose based on the analysis and comparison in Section 5. The surveyed approaches represent a diverse morphology of functionalities demanded by service-oriented simulation. There is a trade-off between using a unified standard framework and permitting diversity with a multi-formalism transformation approach. The compatibility, reusability, and composability of various service resources developed in specific frameworks need further research. The combination of the formalism-based and interoperability protocol based approaches can make full use of the advantages both in theory and practice. Sarjoughian and Zeigler,¹⁷⁴ and Chen et al.¹⁰² investigated the possible combination of traditional DEVS and HLA. However, their combination in a net-centric environment needs further research. In addition, the guidelines of the LCIM can better facilitate the meaningful composition of M&S services. For example, Benali and Bellamine Ben Saoud¹⁷⁵ used the LCIM to combine the ontology-based meta object facility for the knowledge level, BOM for the conceptual level, and HLA for the implementation level. Other emerging techniques, such as Agent⁴⁹ and Cloud computing,¹⁷⁶ can also facilitate new approaches for service-oriented simulation.

To summarize, service-oriented simulation looks very promising for M&S in the information age and net-centric environments. The comprehensive survey and review-derived 3D methodology (or unifying framework) serves as a basic foundation for this field. However, to gain maturity and evolve as a new simulation paradigm requires the continuous joint effort of researchers and practitioners from all communities, in particular academia, industry, and government.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments. The authors would also like to thank Professor Dr Pieter J Mosterman, Dr Justyna Zander,

Professor Dr Feng Yang, Dr Jingjie Li, and Dr Yonglin Lei for their helpful discussions. The authors would like to thank the work of the ‘Three-Dimensional Morphology of Systems Engineering’¹⁴³ and ‘the Levels of Conceptual Interoperability Model’¹⁵¹ team that positively inspired our research.

Some sections of this paper are based on the authors’ Science Letters: Wang WG, Wang WP, Zander J and Zhu YF. Three-dimensional conceptual model for service-oriented simulation. *J Zhejiang Univ Sci A* 2009; 10: 1075-1081. We are granted that ‘detailed research articles can still be published in other journals in the future after Science Letters is published by JZUS (Journal of Zhejiang University SCIENCE A)’. <http://www.zju.edu.cn/jzus/>

Funding

This work was supported by the National Natural Science Foundation of China [grant numbers 60674069, 60574056, 60974073, 60974074].

Conflict of interest statement

None declared.

References

1. Numrich SK, Hieb M and Tolk A. M&S in the GIG environment: an expanded view of distributing simulation. In: *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*. Orlando, FL: Defense Acquisition University Press, 2004.
2. Erl T. *Service-Oriented Architecture: Concepts, Technology, and Design*. Englewood Cliffs, NJ: Prentice Hall PTR, 2005, p.792.
3. Papazoglou MP and van den Heuvel WJ. Service oriented architectures: approaches, technologies and research issues. *Int J Very Large Data Bases* 2007; 16: 389–415.
4. DoD. *Department of Defense Global Information Grid Architectural Vision: Vision for a Net-Centric, Service-Oriented DoD Enterprise Version 1.0*. 2007.
5. Foster I. Service-oriented Science. *Science* 2005; 308: 814–817.
6. Papazoglou MP, Traverso P, Dustdar S and Leymann F. Service-oriented computing: State of the art and research challenges. *Computer* 2007; 40: 38–45.
7. Yang J and Bussler C. Special issue on service oriented modeling – Guest editors’ introduction. *Int J Coop Inf Syst* 2004; 13: 333–335.
8. Brutzman D, Zyda M, Pullen M and Morse KL. Extensible Modeling and Simulation Framework (XMSF) challenges for Web-based modeling and simulation. In: *XMSF 2002 Findings and Recommendations Report: Technical Challenges Workshop and Strategic Opportunities Symposium*, 2002: <http://www.movesinstitute.org/xmsf/XmsfWorkshopSymposiumReportOctober2002.pdf>.
9. Tolk A. A common framework for military M&S and C4I systems. In: *Spring Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2003.
10. Gustavsson PM, Björk Å, Brax C and Planstedt T. Towards service oriented simulations. In: *Fall Simulation Interoperability Workshop*. Orlando, FL: Simulation Interoperability Standards Organization, 2004.
11. Gustavsson PM, Planstedt T, Karlsson G, Brax C and Björk Å. Service oriented simulations concept. In: *10th Simulation Conference and Exhibition SimTecT*. Sydney, Australia, 2005, pp.233–239.
12. Tsai WT. Service-oriented system engineering: a new paradigm. In: *2005 IEEE International Workshop on Service-Oriented System Engineering (SOSE’05)*. Beijing, China, 2005, pp.3–6.
13. Tsai WT, Bai XY and Chen YN. *On service-oriented software engineering*. Beijing, China: Tsinghua University Press, 2008.
14. Karhunen H, Jantti M and Eerola A. Service-oriented software engineering (SOSE) framework. In: Chen J (ed.) *2005 International Conference on Services Systems and Services Management, Vols 1 and 2, Proceedings*. New York: IEEE, 2004, pp.1199–1204.
15. Wang WG, Yu WG, Li Q, Wang WP and Liu XC. Service-Oriented High Level Architecture. In: *European Simulation Interoperability Workshop*. Edinburgh, Scotland: Simulation Interoperability Standards Organization, 2008.
16. DoD. *Department of Defense Net-Centric Services Strategy: Strategy for a Net-Centric, Service Oriented DoD Enterprise*. Washington, D.C, 2007.
17. DoD. *DoDAF V2.0 Volume 1: Introduction, Overview, and Concepts - Manager’s Guide*. Washington, D.C, 2009.
18. Chen Y. Modeling and simulation for and in service-orientated computing paradigm. *Simulation* 2007; 83: 3–6.
19. Gustavson P, Chase T, Root L and Crosson K. Moving towards a Service-Oriented Architecture (SOA) for Distributed Component Simulation Environments. In: *Spring Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2005.
20. Mittal S, Risco-Martin JL and Zeigler BP. DEVS/SOA: A cross-platform framework for net-centric modeling and simulation in DEVS Unified Process. *Simulation* 2009; 85: 419–450.
21. Byrne J, Heavey C and Byrne PJ. A review of Web-based simulation and supporting tools. *Simul Model Pract Theory* 2010; 18: 253–276.
22. Mittal S. *DEVS Unified Process for integrated development and testing of service oriented architectures*. Arizona: The University of Arizona, 2007.
23. Wainer GA, Madhoun R and Al-Zoubi K. Distributed simulation of DEVS and Cell-DEVS models in CD++ using Web-Services. *Simul Model Pract Theory* 2008; 16: 1266–1292.
24. Fan C. *DDSOS: A dynamic distributed service-oriented modeling and simulation framework*. Arizona: Arizona State University, 2006.
25. Li BH, Chai XD, Di YQ, Yu HY, Du ZH and Peng XY. Research on service oriented simulation grid. In: *Proceedings of Autonomous Decentralized Systems (ISADS 2005)*. Chengdu, China, 2005, pp.7–14.

26. Zhang T. *Research on key technologies of service-oriented semantically composable simulation*. Changsha, China: National University of Defense Technology, 2008.
27. Song LL. *Study on SOA-based framework for modeling and simulation and the technologies for simulation service discovery*. Changsha, China: National University of Defense Technology, 2009.
28. NPS. Naval Postgraduate School (NPS) MOVES Institute: Extensible Modeling and Simulation Framework (XMSF), 2004: <http://www.movesinstitute.org/xmsf> (accessed 1 January 2007).
29. Quartel D, Steen MWA, Pokraev S and van Sinderena M. A conceptual framework for service modelling. In: *10th IEEE International Enterprise Distributed Object Computing Conference (EDOC '06)*, 2006, pp.319-330.
30. Balin S and Giard V. A process oriented approach to the service concepts. In: Vincent G (ed.) *2006 International Conference on Service Systems and Service Management*. 2006, pp.785-790.
31. W3C. Web Services Glossary. 2004, <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/> (accessed 16 August 2007)
32. Tolk A, Turnitsa CD, Diallo SY and Winters LS. Composable M&S Web services for net-centric applications. *J Defense Model Sim* 2006; 3: 27-44.
33. Suzić R, Huang Q, Gustavsson PM, Tegnér J and Yi C-H. Core technical framework study: a service framework for distributed simulations in net-centric environments. In: *European Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2009.
34. Zhang T, Liu YS and Zha YB. Semantic Web based simulation service customization and composition. In: *IEEE 7th International Conference on Computer and Information Technology (CIT2007)*. Fukushima, Japan, 2007, pp.235-240.
35. Roberts CA and Dessouky YM. An overview of object-oriented simulation. *Simulation* 1998; 70: 359-368.
36. DoD. *DoD modeling and simulation (M&S) glossary*. United States: Defense Modeling and Simulation Office, 1998, <https://www.dmsomil>.
37. Wallace J, Bizub W, Ceranowicz A, et al. Object model composability and multi-architecture LVC interoperability. In: *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, 2009.
38. IEEE. Systems and software engineering – recommended practice for architectural description of software-intensive systems. *ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15*. 2007, p.c1-24.
39. Zeigler BP, Praehofer H and Kim TG. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, 2nd ed. New York: Academic Press, 2000.
40. Grant T. Unifying planning and control using an OODA-based Architecture. In: *SAICSIT 2005*, 2005, pp.159-170.
41. IEEE. Draft recommended practice for Distributed Simulation Engineering and Execution Process (DSEEP). *IEEE P1730/Dv4.0*, 2009.
42. Yilmaz L and Paspuleti S. Toward a meta-level framework for agent-supported interoperation of defense simulations. *J Defense Model Sim* 2005; 2: 161-175.
43. Yilmaz L. A strategy for improving dynamic composability: Ontology-driven Introspective Agent Architectures. *J Syst, Cyber, Inform* 2007; 5: 1-9.
44. Tolk A, Diallo SY, King RD and Turnitsa CD. *A layered approach to composition and interoperation in complex systems*, SCI 168. Springer, 2009, pp.41-74.
45. Tolk A, Turnitsa CD and Diallo SY. Model-based alignment and orchestration of heterogeneous homeland security applications enabling composition of system of systems. In: *Winter Simulation Conference*, 2007, pp.842-850.
46. Tsai WT, Cao Z, Wei X, Ray P, Huang Q and Sun X. Modeling and simulation in service-oriented software development. *Simulation* 2007; 83: 7-32.
47. Tolk A. What comes after the Semantic Web – PADS implications for the Dynamic Web. In: *20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06)*, 2006, pp.55-62.
48. Tolk A. *Operational modeling & simulation services in military service-oriented architectures-a White Paper summarizing related concepts*. Quantico, VA: 6th Office for Naval Research Workshop on Decision Support, 2004.
49. Yilmaz L and Ören T. *Agent-directed simulation and systems engineering*. Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA, 2009.
50. Tolk A, Diallo SY and Turnitsa CD. Applying the levels of conceptual interoperability model in support of integrability, interoperability, and composability for system-of-systems engineering. *J Syst, Cyber, Inform* 2008; 5: 65-74.
51. Mittal S, Risco-Martín JL and Zeigler BP. DEVSML: automating DEVS execution over SOA towards transparent simulators. In: *DEVS Symposium, Spring Simulation Multiconference*. Norfolk, Virginia, 2007, pp.287-295.
52. Mittal S, Zeigler BP, Risco-Martin JL and Cruz JMdl. WSDL-based DEVS agent for net-centric systems engineering, http://www.acims.arizona.edu/PUBLICATIONS/MittalZeiglerRisco_MAS2008_R2.pdf (2008, accessed 9 October 2008).
53. Mittal S, Risco-Martin JL and Zeigler BP. DEVS-based simulation Web services for net-centric T&E. In: *Summer Computer Simulation Conference SCSC'07*, 2007, pp.357-366.
54. Mittal S and Zeigler BP. DEVS unified process for integrated development and testing of system of systems. In: *Critical Issues in C4I, AFCEA-George Mason University Symposium*, 2008 http://saurabh-mittal.com/downloads/Mittal_Zeigler_DEVS_Unified_Process.pdf (accessed 21 April 2008).
55. Mittal S, Zeigler BP, Risco-Martin JL, Sahin F and Jamshidi M. Chapter 4: Modeling and simulation for systems of systems engineering. In: Jamshidi M (ed.) *System of systems – Innovations for the 21st center*. Wiley & Sons., Inc, 2007.

56. Zeigler BP and Hammonds PE. *Modeling & simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange*. New York: Academic Press, 2007.
57. Sarjoughian H, Kim S, Ramaswamy M and Yau S. A simulation framework for service-oriented computing systems. In: *Winter Simulation Conference*, 2008, pp.845–853.
58. Muqith MA, Sarjoughian HS, Huang D and Yau SS. Simulating adaptive service-oriented software systems. *Simulation* 2010. In press. doi: 10.1177/0037549710382431.
59. Ramaswamy MV. System theory based modeling and simulation of SOA-based software systems. MEng in Modeling and Simulation, Fulton School of Engineering, Arizona State University, Tempe, AZ, 2008, http://www.acims.arizona.edu/PUBLICATIONS/PDF/CSE593_Muthukumar_May2008.pdf (accessed 16 October 2008).
60. Kim S. *Simulator for service-based software system: Design and implementation with DEVS-Suite*. Arizona: The University of Arizona, 2008.
61. Madhoun R. *Web service-based distributed simulation of discrete event models*. Canada: Carleton University, 2006.
62. Al-Zoubi K and Wainer G. Performing distributed simulation with restful web services. In: *Winter Simulation Conference*, 2009, pp.1323–1334.
63. Kim KH and Kang WS. A Web services-based distributed simulation architecture for hierarchical DEVS models. In: Kim TG (ed.) *Artificial intelligence and simulation*. Lecture Notes in Computer Science. Vol. 3397, Springer, 2005, pp.370–379.
64. Hu X, Zeigler BP and Mittal S. Variable structure in DEVS component-based modeling and simulation. *Simulation* 2005; 81: 91–102.
65. Hu X, Zeigler BP, Hwang MH and Mak E. DEVS systems-theory framework for reusable testing of I/O behaviors in Service Oriented Architectures. In: *IEEE International Conference on Information Reuse and Integration*, Las Vegas, NV, 2007, pp.394–399.
66. Kim T, Seo C and Zeigler BP. Web Based Distributed SES/NZER Using Service Oriented Architecture, http://www.acims.arizona.edu/PUBLICATIONS/article_third.pdf (2008, accessed 9 October 2008).
67. Kim T. *Ontology/data engineering based distributed simulation over service oriented architecture for network behavior analysis*. Arizona: The University of Arizona, 2008.
68. Sun XY. *Research on SOA-based runtime framework of DEVS simulation and its key technologies*. Changsha, China: National University of Defense Technology, 2008.
69. Seo C and Zeigler BP. DEVS namespace for Interoperable DEVS/SOA. In: *Winter Simulation Conference*, 2009, pp.1311–1322.
70. Seo C. *Interoperability between DEVS simulators using Service Oriented Architecture and DEVS namespace*. Arizona: The University of Arizona, 2009.
71. Risco-Martin JL, Mittal S, Zeigler BP and Cruz JMDI. From UML Statecharts to DEVS State Machines using XML. In: *Multi-paradigm Modeling, IEEE/ACM International Conference on Modeling Languages and System*, Nashville, 2007.
72. Vangheluwe HLM. *Multi-formalism modelling and simulation*. Ghent University, 2001.
73. Vangheluwe H, Bolduc L and Posse E. DEVS standardization: some thoughts. In: *Winter Simulation Conference*, 2001.
74. Hu X and Zeigler BP. A proposed DEVS standard: model and simulator interfaces, Simulator Protocol (Draft 1.0), 2008, <http://www.acims.arizona.edu/PUBLICATIONS/PDF/GenDEVSSstandardDocument6.pdf> (accessed 3 May 2008).
75. Zeigler B, Mittal S and Hu X. Towards a formal standard for interoperability in M&S/system of systems integration. In: *AFCEA-George Mason University Symposium*, 2008, <http://www.acims.arizona.edu/PUBLICATIONS/PDF/DEVSSstandardGMUAFCEA.pdf> (accessed 18 August 2008).
76. Tsai WT, Fan C, Chen Y and Paul R. DDSOS: a dynamic distributed service-oriented simulation framework. In: *39th Annual Simulation Symposium*, 2006, pp.160–167.
77. Tsai WT, Fan C, Chen Y and Paul R. A service-oriented modeling and simulation framework for rapid development of distributed applications. *Simul Model Pract Theory* 2006; 14: 725–739.
78. Tsai WT, Wei X, Cao Z, Paul R, Chen Y and Xu J. Process specification and modeling language for service-oriented software development. In: *11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS '07)*, 2007, pp.181–188.
79. Jia L and Zhang HM. Research on service oriented distributed M&S framework. *J Syst Simul* 2007; 19: 4680–4684.
80. Tsai WT, Sun X, Huang Q and Karatza H. An ontology-based collaborative service-oriented simulation framework with Microsoft Robotics Studio®. *Simul Model Pract Theory* 2008; 16: 1392–1414.
81. Tsai WT, Chen Y and Fan C. PESOI: Process Embedded Service-Oriented Architecture. *J Software* 2006; 17: 1470–1484.
82. IEEE. IEEE standard for modeling and simulation (M&S) high level architecture (HLA) – framework and rules. *IEEE Std 1516-2000*, 2000, p.i-22.
83. IEEE. IEEE standard for modeling and simulation (M&S) high level architecture (HLA) – object model template (OMT) specification. *IEEE Std 1516.2-2000*, 2001, p.i-130.
84. IEEE. IEEE standard for modeling and simulation (M&S) high level architecture (HLA) – federate interface specification. *IEEE Std 1516.1-2000*, 2001, p.i-467.
85. IEEE. IEEE recommended practice for high level architecture (HLA) federation development and execution process (FEDEP). *IEEE Std 1516.3-2003*, 2003, p.1–32.
86. IEEE. IEEE recommended practice for verification, validation, and accreditation of a federation—an overlay to the high level architecture federation development and execution process. *IEEE Std 1516.4-2007*, 2007.
87. Morse KL, Drake DL and Brunton RPZ. WMDOA integration employing web services for federate communication – an XMSF exemplar. In: *Spring Simulation*

- Interoperability Workshop*. Kissimmee, Florida: Simulation Interoperability Standards Organization, 2003.
88. Morse KL, Drake DL and Brunton RPZ. Web Enabling an RTI – an XMSF Profile. In: *European Simulation Interoperability Workshop*. Stockholm, Sweden: Simulation Interoperability Standards Organization, 2003.
89. Möller B and Löf S. A management overview of the HLA Evolved Web Service API. *Fall Simulation Interoperability Workshop*. Orlando, FL: Simulation Interoperability Standards Organization, 2006.
90. Möller B and Löf S. Mixing service oriented and High Level Architectures in support of the GIG. In: *Spring Simulation Interoperability Workshop*. San Diego, CA: Simulation Interoperability Standards Organization, 2005.
91. Möller B, Morse KL, Lightner M, Little R and Lutz B. HLA Evolved – A Summary of Major Technical Improvements. In: *Fall Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2008.
92. IEEE. IEEE standard for modeling and simulation (M&S) high level architecture (HLA) – framework and rules. *IEEE Std 1516-2010*, 2010.
93. IEEE. IEEE standard for modeling and simulation (M&S) high level architecture (HLA) – federate interface specification. *IEEE Std 1516.1-2010*, 2010.
94. IEEE. IEEE standard for modeling and simulation (M&S) high level architecture (HLA) – object model template (OMT) specification. *IEEE Std 1516.2-2010*, 2010.
95. Pitch. Pitch Web Services Connector, 2007, <http://www.pitch.se/products/web-services-connector/pitch-web-services-connector.html> (accessed 4 December 2007).
96. MAK. The MAK RTI on the Road to HLA Evolved and Reaching Beyond 1.3 and 1516, 2008, <http://www.mak.com> (accessed 29 May 2008).
97. MAK. MAK RTI 3.2 release notes, 2007, <http://www.mak.com> (accessed 12 May 2008).
98. Pan K, Turner SJ, Cai W and Li Z. Implementation of data distribution management services in a service oriented HLA RTI. In: *Winter Simulation Conference*. 2009, pp.1027–1038.
99. Tang S, Xiao T and Fan W. A collaborative platform for complex product design with an extended HLA integration architecture. *Simul Model Pract Theory* 2010; 18: 1048–1068.
100. Zhang H. A solution of multidisciplinary collaborative simulation for complex engineering systems in a distributed heterogeneous environment. *Sci China Series F: Inform Sci* 2009; 52: 1848–1862.
101. SISO. Base Object Model (BOM) template specification. *SISO-STD-003-2006*, 2006.
102. Chen B, Zhong R and Vangheluwe H. Integrating base object model components into DEVS-based simulation. *J Defense Model Sim* 2010; 7: 241–256.
103. Wang WG, Xu YP, Chen X, Li Q and Wang WP. High Level Architecture Evolved Modular Federation Object Model. *J Syst Eng Electron* 2009; 20: 625–635.
104. Möller B, Löfstrand B and Karlsson M. An overview of the HLA evolved modular FOMs. In: *Spring Simulation Interoperability Workshop*. Norfolk, VA: Simulation Interoperability Standards Organization, 2007.
105. Möller B, Antelius F, Johansson M, Löfstrand B and Wihlborg Å. Processes and tools for management and reuse of FOM modules. In: *Fall Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2010.
106. Smith RD. Simulation as an IT service. In: *European Simulation Interoperability Workshop*. Genoa, Italy: Simulation Interoperability Standards Organization, 2007.
107. Scudder R. High Level Architecture (HLA) Evolved PDG – agenda, overview. In: *Fall Simulation Interoperability Workshop*. Orlando, FL: Simulation Interoperability Standards Organization, 2006.
108. Möller B and Karlsson M. Developing well-balanced federations using the HLA Evolved Smart Update Rate Reduction. In: *Fall Simulation Interoperability Workshop*. Orlando, FL: Simulation Interoperability Standards Organization, 2005.
109. Möller B, Karlsson M and Löfstrand B. Developing fault tolerant federations using HLA Evolved. In: *Spring Simulation Interoperability Workshop*. San Diego, CA: Simulation Interoperability Standards Organization, 2005.
110. Möller B, Dahlin C and Karlsson M. Developing Web centric federates and federations using the HLA Evolved Web Services API. In: *Spring Simulation Interoperability Workshop*. Norfolk, VA: Simulation Interoperability Standards Organization, 2007.
111. Straßburger S, Schulze T and Fujimoto R. Peer Study Final Report: Future Trends in Distributed Simulation and Distributed Virtual Environments, 2008, http://www.sisostds.org/index.php?tg=fileman&idx=get&id=5&gr=Y&path=Standards+Activity+Committee%2FDocuments+for+Review&file=PeerStudy_DS_DVE_v1.0.pdf.
112. Tolk A, Turnitsa C and Diallo S. Implied ontological representation within the levels of conceptual interoperability model. *Intell Decis Technol* 2008; 2: 3–19.
113. He M, Qiu HP and Liu XM. Research of modeling and simulation on XMSF. *J Syst Simul* 2005; 17: 1032–1036.
114. Tan J, Li BH and Chai XD. Technology research of extensible modeling and simulation framework – XMSF. *J Syst Simul* 2006; 18: 96–101.
115. SAIC. HLA Integration Employing Web Services for Federate Communications The Extensible Modeling & Simulation Framework (XMSF), <http://www.movesinstitute.org/xmsf> (accessed 1 February 2007).
116. Buss A and Ruck J. Joint modeling and analysis using XMSF Web Services. In: *Winter Simulation Conference*, 2004, pp.1032-1038.
117. Sohn MM. Advanced M&S framework based on autonomous Web service in Korea: Intelligent-XMSF approach. *Systems modeling and simulation: theory and applications*. Berlin: Springer, 2005, pp.150–158.
118. SISO. SISO Discussions, 2005, <http://discussions.sisostds.org> (accessed October 2007).

119. Clery D. Infrastructure – Can grid computing help us work together? *Science* 2006; 313: 433–434.
120. Yu J and Han YB. *Service-oriented computing: Principles and application*. Beijing, China: Tsinghua University Press, 2006.
121. Foster I, Frey J, Graham S, et al. *Modeling stateful resources with Web Services*. Globus Alliance, 2004, <http://www.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf> (accessed 1 December 2008).
122. Chen X, Cai W, Turner SJ and Wang Y. SOAr-DSGrid: Service-Oriented Architecture for Distributed Simulation on the Grid. In: *20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06)*, 2006, pp.65–73.
123. Rycerz K, Bubak M, Malawski M and Sloot P. A framework for HLA-based interactive simulations on the grid. *Simulation* 2005; 81: 67–76.
124. Rycerz KJ. *Grid-based HLA simulation support*. University van Amsterdam and AGH Krakow, 2006.
125. Pan K, Turner SJ, Cai W and Li Z. A Service Oriented HLA RTI on the Grid. In: *2007 IEEE International Conference on Web Services (ICWS 2007)*, 2007, pp.984–992.
126. Xie Y, Teo YM, Cai W and Turner SJ. Service provisioning for HLA-based distributed simulation on the Grid. In: *19th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation*. Monterey, USA: IEEE Computer Society Press, 2005, pp.282–291.
127. Zhang W and Zhang YB. Survey of Grid-enabled advanced distributed simulation. *J Syst Simul* 2008; 20: 1089–1093.
128. Zhang W. *Research on key technologies of service grid for parallel and distributed simulations*. Changsha, China: National University of Defense Technology, 2009.
129. Chai X, Yu H, Du Z, Hou B and Li BH. Research and application on service oriented infrastructure for networkitized M&S. In: *Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops*, 2006, p.67.
130. Li BH, Chai XD, Hou BC, et al. New distributed collaborative simulation system – simulation grid. *J Syst Simul* 2008; 20: 5423–5430.
131. Strelch TP, Adams DP and Sloan WW. Simulation-based transformation with the service integration/interoperation infrastructure. *Technol Rev J* 2005; 13: 99–115.
132. Strelch T, Cluff C and Johansson UM. Future-proof simulation integration/interoperation infrastructure. In: *Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, Orlando, FL, 2001.
133. ROLANDS. JTLS Executive Overview 3.1, 2006, <http://www.rolands.com> (accessed 17 January 2007).
134. World Wide Web Consortium, <http://www.w3.org/> (accessed 16 August 2007)
135. Biswas D. *Hierarchical Web services compositions: Visibility, compensation and monitoring*. Canada: Memorial University of Newfoundland, 2005.
136. Cao F. *Model driven development and dynamic composition of Web services*. Alabama: The University of Alabama at Birmingham, 2005.
137. Deng SG. *Research on automatic service composition and formal verification*. Zhejiang, China: Zhejiang University, 2007.
138. Yang YP. *Research on automatic service composition and its key technologies*. Changsha, China: National University of Defense Technology, 2007.
139. Hendler J. Communication – science and the Semantic Web. *Science* 2003; 299: 520–521.
140. Lee K and Kim B. A dynamic defense modeling and simulation methodology using Semantic Web Services. In: *International Conference on Modeling, Simulation & Visualization Methods (MSV)*, 2009.
141. Hu J and Zhang H. Ontology based collaborative simulation framework using HLA & Web Services. In: *Spring Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2009.
142. Ritchey T. Problem structuring using computer-aided morphological analysis. *J Operat Res Soc* 2006; 57: 792–801.
143. Hall AD. Three-dimensional morphology of systems engineering. *IEEE Tran Syst Sci Cyber* 1969; 5: 156–160.
144. Waite B. Modeling and Simulation (M&S) Body of Knowledge (BoK) for Academic Night. In: *Spring Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2006.
145. Jamshidi M. *System of systems engineering—Innovations for the 21st Century*. New York: John Wiley & Sons, 2009.
146. Petty MD and Weisel EW. A composability lexicon. In: *Spring Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization, 2003.
147. Ford TC, Colombi JM, Graham SR and Jacques DR. A survey on interoperability measurement. In: *Twelfth International Command and Control Research and Technology Symposium (12th ICCRTS)*, 19–21 June 2007, Newport, RI, 2007.
148. Davis PK and Anderson RH. *Improving the composability of Department Of Defense models and simulations*. Santa Monica, 2002.
149. Petty MD, Weisel EW and Mielke RR. A formal approach to composability. In: *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, 2003.
150. Weisel EW. *Models, composability, and validity*. Virginia: Old Dominion University, 2004.
151. Tolk A and Muguira JA. The levels of conceptual interoperability model. In: *Fall Simulation Interoperability Workshop*. Orlando, FL: Simulation Interoperability Standards Organization, 2003.
152. Yilmaz L. On the need for contextualized introspective models to improve reuse and composability of defense simulations. *J Defense Model Sim* 2004; 1: 141–151.
153. Yilmaz L and Ören TI. A conceptual model for reusable simulations within a model-simulator-context framework. In: *Conceptual Modeling and Simulation Conference at the First International Mediterranean*

- Modeling Multiconference*. Genoa, Italy, 2004, pp.235–241.
154. Fishwick PA. *Handbook of dynamic system modeling*. USA: Taylor & Francis Group, 2007.
 155. Hofmann MA. Challenges of model interoperability in military simulations. *Simulation* 2004; 80: 659–667.
 156. Yue K, Wang XL and Zhou AY. Underlying techniques for Web services: A survey. *J Software* 2004; 15: 428–442.
 157. Yu WD, Radhakrishna RB, Pingali S and Kolluri V. Modeling the measurements of QoS requirements in Web service systems. *Simulation* 2007; 83: 75–91.
 158. McKenzie FD, Petty MD and Xu Q. Usefulness of software architecture description languages for modeling and analysis of federates and federation architectures. *Simulation* 2004; 80: 559–576.
 159. Robinson S. Modes of simulation practice: approaches to business and military simulation. *Simul Model Pract Theory* 2002; 10: 513–523.
 160. Robinson S. Distributed simulation and simulation practice. *Simulation* 2005; 81: 5–13.
 161. Oliver DW. Systems engineering and software engineering, contrasts and synergism. In: *The Seventh International Workshop on Computer-Aided Software Engineering*, 1995, pp.186–193.
 162. IEEE. Systems and software engineering – system life cycle processes. *IEEE Std 15288-2008*, 2008, pp.1-84.
 163. Mei H and Shen JR. Progress of research on software architecture. *J Software* 2006; 17: 1257–1275.
 164. Hitchins DK. *Systems engineering: A 21st Century systems methodology*. Great Britain: John Wiley & Sons, 2007.
 165. Wang W, Tolk A and Wang W. The levels of conceptual interoperability model: applying systems engineering principles to M&S. In: *SCS Spring Simulation Multiconference (SpringSim'09)*. San Diego, CA: ACM, 2009, pp.375–384.
 166. Wang WG, Wang WP, Zander J and Zhu YF. Three-dimensional conceptual model for service-oriented simulation. *J Zhejiang Univ Sci A* 2009; 10: 1075–1081.
 167. Li BH, Chai XD, Zhu WH, et al. Some focusing points in development of modern modeling and simulation technology. *J Syst Simul* 2004; 16: 1871–1878.
 168. Yilmaz L, Davis P, Fishwick PA, et al. Panel discussion: What makes good research in modeling and simulation: sustaining the growth and vitality of the M&S discipline. In: *Winter Simulation Conference*, 2008, pp.677–688.
 169. Liu L, Russell D, Xu J, et al. Modelling and simulation of network enabled capability on service-oriented architecture. *Simul Model Pract Theory* 2009; 17: 1430–1442.
 170. Balci O, Arthur JD and Nance RE. Accomplishing Reuse with a Simulation Conceptual Model. In: *Winter Simulation Conference*, 2008, pp.959–965.
 171. Blais C, Brutzman D, Drake D, et al. Extensible Modeling and Simulation Framework (XMSF) 2004. Project Summary Report. Monterey, CA, 2004.
 172. Taylor SJE, Turner SJ and Strassburger S. Guidelines for Commercial Off-the-Shelf simulation package interoperability. In: *Winter Simulation Conference*, 2008, pp.193–204.
 173. Strassburger S, Schulze T and Fujimoto R. Future trends in distributed simulation and distributed virtual environments: results of a peer study. In: *Winter Simulation Conference*, 2008, pp.777–785.
 174. Sarjoughian HS and Zeigler BP. DEVS and HLA: Complementary paradigms for modeling & simulation? *Trans Soc Comput Simul* 2000; 17: 187–197.
 175. Benali H and Bellamine Ben Saoud Ns. Towards a component-based framework for interoperability and composability in Modeling and Simulation. *Simulation* 2010: In press. doi: 10.1177/0037549710373910.
 176. Buyya R, Ranjan R and Calheiros RN. Modeling and simulation of scalable cloud computing environments and the CloudSim Toolkit: challenges and opportunities. In: *7th High Performance Computing and Simulation (HPCS 2009) Conference*, Leipzig, Germany, 2009.
- Wenguang Wang** is a PhD candidate at College of Information Systems and Management, National University of Defense Technology (NUDT), China. He is a member of the SCS, SISO, and CASS (Chinese Association for System Simulation). He is also an invited reviewer for the international journal *Simulation Modelling Practice and Theory*. His research interests include service-oriented simulation, High Level Architecture, DEVS, simulation composability and interoperability, etc. Email: wgwangnudt@gmail.com
- Weiping Wang** is a professor at College of Information Systems and Management, National University of Defense Technology (NUDT), China. He is the vice dean of the Graduate School, NUDT. He is the founder and director of Systems Simulation Lab, NUDT. He has over 20 years of experience in systems modeling and simulation community. He is a PhD supervisor. His research interests include systems simulation, system of systems engineering, systems of systems simulation, simulation based acquisition, simulation composability and interoperability, etc.
- Yifan Zhu** is a professor at College of Information Systems and Management, National University of Defense Technology (NUDT), China. He is the deputy director of the Department of Systems Engineering, College of Information Systems and Management, NUDT. He is the co-founder of Systems Simulation Lab, NUDT. He has over 20 years of experience in systems modeling and simulation community. He is a PhD supervisor. He was a visiting scholar in College of Engineering, Virginia Tech University, USA from 2007 to 2008. His research interests include systems simulation, virtual prototyping, simulation based acquisition, simulation composability and interoperability, etc. He is the principal investigator of the National Natural

Science Foundation of China project on ‘Composable Simulation’ under grant number 60574056.

Qun Li is a professor at College of Information Systems and Management, National University of Defense Technology (NUDT), China. He is currently the director of Systems Simulation Lab, NUDT. He has nearly 20 years of experience in systems

modeling and simulation community. He is a MSc supervisor. His research interests include systems simulation, simulation based acquisition, service-oriented simulation, simulation composability and interoperability, etc. He is the principal investigator of the National Natural Science Foundation of China project on ‘Service-oriented Simulation’ under grant number 60674069.

Appendix

Table A1. Comparison of frameworks from M&S dimension (ID view)

Frameworks	Experimental Frame/Pragmatics	Model/Semantics	Simulator/Syntax
Formalism based	DUNIP testing models/ federations	DEVSML, DEVS, SES	DEVS simulation protocol, DEVS/SOA
Model driven	System/environment agent services	PSML	HLA-based
Interoperability protocol based	N/A	BOM, SOM, FOM, modular SOM/FOM	RTI, Web service API
XMSF	N/A	N/A	N/A
OGSA based	N/A	Model resources	Simulator functions and resources
Others	N/A	Model services, ontologies	Simulator services, ontologies

The increasing gray intensity of the cells identifies nice-to-have and core issues, respectively.

Table A2. Comparison of frameworks from service-orientation dimension (ID view)

Frameworks	Broker	Requester	Provider	Transport	Messaging	Description	P&D	Composition	QoS
Formalism based	F2	F2	F1, F2, F3	F1, F2, F3	F1, F2, F3	F1, F2, F3	F2	F2	F2, F3
Model driven	M1	M1	M1	M1	M1	M1	M1	M1	M1
Interoperability protocol based			II	II	II	II			
XMSF			XI	XI	XI	XI			
OGSA based			G1	G1	G1	G1			G1
Others	O1, O2		O1, O2	O1, O2	O1, O2	O1, O2	O1, O2	O1, O2	O2

The increasing gray intensity of the cells identifies nice-to-have, supporting, and core issues, respectively. F1 = DUNIP; F2 = SOAD; F3 = D-CD++; M1 = DDSOS; II = SOHLA; XI = XMSF; G1 = OGSA based frameworks; O1 = Si3; O2 = Ontology driven frameworks; P&D = Publication and Discovery.

Table A3. Comparison of frameworks from engineering dimension (ID view)

Frameworks	Requirements	Design	Implementation	Testing	Deployment	Post-development
Formalism based	Y	Y	Y	Y	Y	
Model driven		Y	Y	Y	Y	Y
Interoperability protocol based		Y	Y	Y		
XMSF		Y	Y			
OGSA based		Y	Y			
Others		Y	Y			

The increasing gray intensity of the cells identifies supporting and core issues, respectively.

Table A4. Comparison of frameworks from narrow service-oriented simulation (M&S vs. Services 2D view)

	Broker	Requester	Provider	Transport	Messaging	Description	P&D	Composition	QoS
EF/P	M1	M1	M1		F1	F1, F3, M1		F1(SES)	
M/Se	F2 ^T , M1, M1 ^T , G1, O2	F1 (User), F2 ^T , M1, M1 ^T	F1 (compile, transform, validate), F2 ^T , M1, M1 ^T , G1, O1	F2 ^T (hardware and software)	F2 ^T , M1, M1 ^T , G1	(DEVSM-L), F2 ^T , M1, M1 ^T , G1, O2	F2 ^T , O2	F1(SES, static), F2 ^T (static), M1, M1 ^T , O2	F2 ^T , M1, G1, O2
S/Sy	G1, O2	F1 (User),	F1 (DEVS/ SOA), F3, M1, I1, X1, G1, O1	F3, I1, X1	F1, F3, M1, I1, X1, G1	F1, F3, M1, I1, X1, G1, O2	M1, O2	F1(SES, static), M1, O2	F3, M1, G1, O2

The increasing gray intensity of the cells identifies nice-to-have, supporting, and core issues, respectively. Elements marked with a superscript 'T' (transposition) identify M&S for SOAs; normal elements identify SOAs for M&S. EF/P = Experimental Frame/Pragmatics; M/Se = Model/Semantics; S/Sy = Simulator/Syntax; F1 = DUNIP; F2 = SOAD; F3 = D-CD++; M1 = DDSOS; I1 = SOHLA; X1 = XMSF; G1 = OGSA based frameworks; O1 = Si3; O2 = Ontology driven frameworks; EF = Experimental Frame; SES = System Entity Structure; DEVSM-L = DEVS Modeling Language; P&D = Publication and Discovery.

Table A5. Comparison of frameworks from M&S engineering (M&S vs. Engineering 2D view)

	Requirements	Design	Implementation	Testing	Deployment	Post-development
EF/P	F1	F1, F2 ^T , F3, M1	F1, F2 ^T , F3, M1		F1	
M/Se	F1, M1(PSML), O2	F1, F2 ^T , M1(PSML), G1, O1	F1, F2 ^T , M1(PSML), G1, O1	F1, M1	F1, M1	M1, O2
S/Sy	I1, O2	F1, F3, M1, I1, G1, O1	F1, F3, M1, I1, G1, O1	I1	F1, I1	F3 (Performance), G1, O2

The increasing gray intensity of the cells identifies nice-to-have, supporting, and core issues, respectively. Elements marked with a superscript 'T' (transposition) identify M&S for SOAs; normal elements identify SOAs for M&S. EF/P = Experimental Frame/Pragmatics; M/Se = Model/Semantics; S/Sy = Simulator/Syntax; F1 = DUNIP; F2 = SOAD; F3 = D-CD++; M1 = DDSOS; I1 = SOHLA; G1 = OGSA based frameworks; O1 = Si3; O2 = Ontology driven frameworks.

Table A6. Comparison of frameworks from service-oriented engineering (Services vs. Engineering 2D view)

	Broker	Requester	Provider	Transport	Messaging	Description	P&D	Composition	QoS
Req		F1	F1	F2 ^T	F1	F1		M1	
Dsn	F2 ^T , M1, O2	F1, F2 ^T	F1, F2 ^T , F3, M1, I1, G1, O1	F1, F2 ^T , M1, I1, G1	F1, F2 ^T , F3, M1, I1, G1	F1, F2 ^T , F3, M1, I1, G1, O1, O2	F2 ^T , M1, O2	F1(SES, static), F2 ^T (static), M1, O2	F2 ^T , M1, G1, O2
Imp	F2 ^T , M1, O2	F1, F2 ^T	F1, F2 ^T , F3, M1, I1, G1, O1	F1, F2 ^T , M1, I1, G1	F1, F2 ^T , F3, M1, I1, G1	F1, F2 ^T , F3, M1, I1, G1, O1, O2	F2 ^T , M1, O2	F1(SES, static), F2 ^T (static), M1, O2	F2 ^T , M1, G1, O2
Tst			F1, M1, I1			F1, M1		M1	
Dply		F1	F1, M1	F1	F1	F1			
PstD			M1						

The increasing gray intensity of the cells identifies nice-to-have, supporting, and core issues, respectively. Elements marked with a superscript 'T' (transposition) identify M&S for SOAs; normal elements identify SOAs for M&S. F1 = DUNIP; F2 = SOAD; F3 = D-CD++; M1 = DDSOS; I1 = SOHLA; G1 = OGSA based frameworks; O1 = Si3; O2 = Ontology driven frameworks; Req = Requirement; Dsn = Design; Imp = Implementation; Tst = Test; Dply = Deployment; PstD = Post-development; P&D = Publication and Discovery.

Table A7. Comparison of frameworks from model/semantics's perspective (3D view)

	Broker	Requester	Provider	Transport	Messaging	Description	P&D	Composition	QoS
Req			FI	F2 ^T	FI	FI, MI			
Dsn	F2 ^T , O2	FI (User), F2 ^T , MI (PSML ^T)	FI(compile, transform, validate), F2 ^T , MI(PSML ^T), GI, OI	FI, F2 ^T , MI(PSML ^T)	FI, F2 ^T , MI(PSML ^T)	FI(DEVSMML), F2 ^T , MI(PSML ^T), OI, O2	F2 ^T , O2	FI(SES, static), F2 ^T (static), MI(PSML ^T), O2	F2 ^T , O2
Imp	F2 ^T , O2	FI (User), F2 ^T , MI (PSML ^T)	FI(compile, transform, validate), F2 ^T , MI(PSML ^T), GI, OI	FI, F2 ^T , MI(PSML ^T)	FI, F2 ^T , MI(PSML ^T)	FI(DEVSMML), F2 ^T , MI(PSML ^T), OI, O2	F2 ^T , O2	FI(SES, static), F2 ^T (static), MI(PSML ^T), O2	F2 ^T , O2
Tst			FI, MI			FI			MI
Dply			FI, MI	FI, MI	FI, MI	FI, MI		MI	MI
PstD			MI	MI	MI	MI		MI	MI

The increasing gray intensity of the cells identifies nice-to-have, supporting, and core issues, respectively. Elements marked with a superscript 'T' (transposition) identify M&S for SOAs; normal elements identify SOAs for M&S. FI = DUNIP; F2 = SOAD; MI = DDSOS; GI = OGSA based frameworks; OI = Si3; O2 = Ontology driven frameworks; SES = System Entity Structure; DEVSMML = DEVS Modeling Language; Req = Requirement; Dsn = Design; Imp = Implementation; Tst = Test; Dply = Deployment; PstD = Post-development; P&D = Publication and Discovery.

Table A8. Comparison of frameworks from simulator/syntax's perspective (3D view)

	Broker	Requester	Provider	Transport	Messaging	Description	P&D	Composition	QoS
Req					FI				
Dsn	MI, O2	FI (User), MI	FI, F3, MI, XI, II, GI, OI	FI, MI, XI, II, GI	FI, F3, MI, XI, II, GI	FI, F3, MI, XI, II, GI, OI, O2	MI, O2	MI, O2	MI, O2
Imp	MI, O2	FI (User), MI	FI, F3, MI, XI, II, GI, OI	FI, MI, XI, II, GI	FI, F3, MI, XI, II, GI	FI, F3, MI, XI, II, GI, OI, O2	MI, O2	MI, O2	F3, MI, O2
Tst			II						
Dply			FI, MI	FI, MI	FI, MI	FI, MI		MI	MI
PstD			MI	MI	MI	MI		MI	MI

The increasing gray intensity of the cells identifies nice-to-have, supporting, and core issues, respectively. FI = DUNIP; F3 = D-CD++; MI = DDSOS; II = SOHLA; XI = XMSF; GI = OGSA based frameworks; OI = Si3; O2 = Ontology driven frameworks; Req = Requirement; Dsn = Design; Imp = Implementation; Tst = Test; Dply = Deployment; PstD = Post-development; P&D = Publication and Discovery.

Table A9. Comparison of frameworks from experimental frame/pragmatics's perspective (3D view)

	Broker	Requester	Provider	Transport	Messaging	Description	P&D	Composition	QoS
Req						FI			
Dsn	O2		MI	FI, MI	FI, MI	FI, MI	O2	MI	F2 ^T , F3, MI, GI, O2
Imp	O2		MI	FI, MI	FI, MI	FI, MI	O2	MI	F2 ^T , F3, MI, GI, O2
Tst									
Dply				FI	FI	FI			
PstD									

The gray intensity of the cells identifies nice-to-have issues. Elements marked with a superscript 'T' (transposition) identify M&S for SOAs; normal elements identify SOAs for M&S. FI = DUNIP; F2 = SOAD; F3 = D-CD++; MI = DDSOS; GI = OGSA based frameworks; O2 = Ontology driven frameworks; Req = Requirement; Dsn = Design; Imp = Implementation; Tst = Test; Dply = Deployment; PstD = Post-development; P&D = Publication and Discovery.