Ontological, Epistemological, and Teleological Perspectives on Service-Oriented Simulation Frameworks

Wenguang Wang, Weiping Wang, Qun Li, and Feng Yang

National University of Defense Technology
Changsha, China

Abstract. This chapter investigates service-oriented simulation frameworks from the ontological, epistemological, and teleological perspectives. First, we give an overview of various specific frameworks that imply particular referential ontological, epistemological, and teleological perspectives for real world systems. Then we combine the partial considerations derived from the review into a unifying framework. It inspects the crossover between the disciplines of M&S, service-orientation, and software/systems engineering. From a methodological perspective, we show its ontological, epistemological, and teleological implications for abstract approaches. The unifying framework can, in turn, facilitate the classification, evaluation, selection, description, and prescription of the known or proposed frameworks. Thus, the referential and methodological perspectives build a systematical philosophical foundation of the service-oriented simulation paradigm.

Keywords: Ontology, Epistemology, Teleology, Service-oriented simulation, Service-oriented architecture (SOA), Software engineering, Systems engineering, Referentiality, Methodology, Composability, Interoperability.

1 Introduction

With the prevalence of net-centric environments, the modeling and simulation (M&S) community is highly demanded to offer agile capabilities (e.g. for intelligent applications) by providing, reusing, and composing heterogeneous resources. Service-Oriented Architectures (SOA) [1] as well as its implementation techniques (Web Services etc.) provides such an opportunity. Therefore, the use of SOA to extend the capabilities of M&S frameworks has attracted increasing attention [2]. Various service-oriented simulation frameworks have been proposed or implemented by different institutes using different formalisms and techniques. These include formalism-based [3], model-driven [4], interoperability protocol based [5], Open Grid Services Architecture (OGSA) based [6], and ontology driven

A. Tolk (Ed.): Ontology, Epistemology, & Teleology for Model. & Simulation, ISRL 44, pp. 335–358.
springerlink.com

[7] frameworks, as well as the eXtensible Modeling and Simulation Framework (XMSF) [8]. All of these imply particular ontological, epistemological, and teleological considerations that direct and shape the diversities of implementations by developers.

Ontology, epistemology, and teleology build the philosophical foundation of a discipline [9,10]. Ontology is the study of what exists, often captured as a finite set of concepts and their relations. Epistemology focuses on the way we define knowledge, especially how we come to know new knowledge. Teleology emphasizes on the study of purpose and action, i.e. purposeful behavior while seeking of a goal. To gain maturity and evolve as a new simulation paradigm/ discipline, service-oriented simulations must build a solid philosophical foundation from the ontological, epistemological, and teleological perspectives. Therefore, we must study (1) the purpose of the paradigm/discipline (teleology); (2) the philosophical foundations and the basic concepts/elements thereof (ontology); (3) its formation and evolution (epistemology); (4) a systematic methodology that include the formation, way of thinking, and assessment of particular methods (epistemology and teleology); (5) ways to find and solve new problems/gaps (epistemology and teleological activities).

Two key challenges still need to be addressed in the research of service-oriented simulation. First, developers of each framework use particular ontological, epistemological, and teleological assumptions that lead to varieties of formalisms, designs, techniques, and implementations. It is thus necessary to undertake a review to facilitate the classification, evaluation, and selection of the reviewed or future frameworks. Second, to the best of our knowledge, there is no efficient way to connect and compare one framework to another. Therefore a general (or high level) systematic philosophical foundation of the paradigm derived from the partial perspectives is needed.

This chapter has two interrelated goals. The first is to undertake an overview of various service-oriented simulation approaches that imply particular ontological, epistemological, and teleological considerations. The second goal is to combine the partial considerations derived from the state-of-the-art into one unifying framework that reflect a systematical philosophical foundation of the service-oriented simulation discipline. Ontology, epistemology, and teleology have both referential (for real world) and methodological (for abstract methods) categories [11]. In this chapter they exhibit referential properties in the particular approaches, while reveal methodological characteristics in the unifying framework.

The rest of the chapter is organized as follows. In Section 2, we give a comprehensive survey of various service-oriented frameworks. Deriving from the review, we propose a novel unifying methodology in Section 3 and show its ontological and epistemological implications. Driven by teleology, we use the unifying frame to describe, compare, and prescribe the reviewed approaches in Section 4. In the last section we describe the contributions and recommend future work.

2 Particular Ontological, Epistemological, and Teleological Perspectives on Specific Frameworks

Different people have different perceptions, understandings, and assumptions of the service-oriented simulation paradigm. Thus, the diversity of their ontological, epistemological, and teleological perspectives leads to various design and implementations of the reviewed frameworks.

2.1 Formalism-Based Framework

The category has some common ontological and epistemological assumptions. They all depend on certain formalisms in a theoretical or mathematical way e.g. the Discrete Event System Specification (DEVS). More or less, they cover the domain ontologies of M&S (referent, model, simulator, and experimental frame), service-orientation, and software/system engineering.

(1) DEVS Unified Process framework (DUNIP)

From the teleological perspective, DUNIP was proposed by Mittal and his colleagues [3] for the integrated development and testing of service-oriented architectures. Additionally, the applied projects [3] such as the Joint Close Air Support (JCAS) Model, DoDAF-based Activity Scenario have more real world teleological characteristics. From the ontological and epistemological viewpoints, the authors use DEVS as a unified model specification, take simulator as services while models as resources, and propose a bifurcated model-continuity methodology for system engineering.

(2) DEVS framework for service-oriented computing systems (SOAD)

From the teleological perspective, SOAD [12] was proposed to extend the DEVS with basic SOA concepts for modeling and simulation of service-oriented computing systems. 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. From the ontological and epistemological viewpoints, the conceptual framework of an SOAD is reported in [12]. The research on SOAD concerns the three roles of a SOA, messaging patterns, primitive and composite service composition, and hardware models for router links.

(3) Web services based Cell-DEVS framework (D-CD++)

From the teleological perspective, The D-CD++ [13] is proposed to Web-enable the DEVS formalism that defines spatial models as cell spaces. From the ontological and epistemological viewpoints, the set of service interfaces in D-CD++ includes session management, configuration, simulation modeling and control, and retrieving data interfaces. The execution of D-CD++ conforms to parallel DEVS simulation protocols and adopts a global conservative time management strategy.

2.2 Model-Driven Framework

From the ontological and teleological perspective, 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) [4, 14] is a typical example. From the ontological and epistemological viewpoints, they define a Process Specification and Modeling Language for Services (PSML-S) to model SOA systems. They hold that dynamic rebinding, re-composition, and re-architecture are the merits of the framework. Therefore some agent services and mechanisms are proposed to support these characteristics. Service-oriented systems engineering (SOSE) and MDA provide whole lifecycle support.

2.3 Interoperability Protocol Based Framework

From the ontological and teleological perspective, this approach utilizes the some interoperability protocols (e.g., HLA) as the standard simulation bus for service integration and information exchange. A typical example is service-oriented HLA (SOHLA) [5]. From the epistemological perspective, researchers of this category suggest to web enable the HLA at four layers, i.e. communication layer (such as Web-Enabled RTI), interface specification layer (e.g., HLA Evolved Web Service API and Unified Architecture [15]), federate interface layer (such as the HLA Connector [15]) and the application layer (e.g., HLA Island). The BOM and modular FOM [16] can facilitate model interoperations, and the FEDEP can provide a system engineering basis. A recent PhD dissertation is reported by Wang [17] that aims to improve the service ability and composability of the HLA framework based on some unifying theories.

2.4 EXtensible Modeling and Simulation Framework (XMSF)

From the ontological and teleological perspective, XMSF [18] is defined as a composable set of standards, profiles, and recommended practices for Web-based M&S. The practice of XMSF includes the Web-Enabled RTI and the project using XMSF to connect Navy Simulation Systems, Simkit, and CombatXXI, for joint modeling and analysis sponsored by SAIC. From the ontological and epistemological perspective, Web/XML, Internet/Networking and M&S are regarded as the major focus areas of XMSF.

2.5 Open Grid Services Architecture Based Framework (OGSA)

From the ontological and teleological perspective, 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. Grids can be classified into computing, storage, data, knowledge, and service Grids according to the properties of the resources at the nodes. From the epistemological perspective, researchers and practitioners regard the HLA/RTI services/components as Grid services. These frameworks include the Cosim-Grid [6], SOAr-DSGrid [19], G-HLAM [20], and SOHR [21].

2.6 Other Service-Oriented Simulation Frameworks

Besides the above classical frameworks, Northrop Grumman's Service Integration/Interoperation Infrastructure (Si3) [22] was proposed to support simulation-based transformation. Ontology-driven framework [7,23,24] uses ontologies or semantic Web to improve the communication between users and Web services that use different terminologies.

2.7 A Summary and Overall Comparison

The reviewed approaches exhibit common M&S, Service-orientation, and software/system engineering ontologies more or less. We give an overall comparison in Table 1. We compare the six reviewed categories of approaches, listed as rows, with respect to the metrics for typical examples and three important dimensions. The advantages and limitations of each framework are listed in Table 2. 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.

3 Ontological and Epistemological Perspectives on a Unifying Framework

Based on the partial ontological, epistemological, and teleological assumptions that lead to the ad-hoc frameworks, this section proposes a unifying framework or methodology (i.e., a three-dimensional reference model) derived from the review. It has upper ontological, epistemological, and teleological characteristics compared with all the specific methods. It also reveals the common functionalities and totality of research issues in the service-oriented simulation paradigm.

3.1 Principle of the Unifying Methodology

Based on our previous detailed review [25] and the explanations from the ontological, epistemological, and teleological viewpoints above, we identify (at least) three distinct, yet related fundamental dimensions (domains or viewpoints): M&S, service-orientation, and software/systems engineering. We regard the three dimensions as independent or orthogonal conceptual domains (sub-ontologies), since each has its own relatively complete and mature set of theory, approaches, standards, techniques, practices, and applications.

Table 1 Overall comparison of classical service-oriented simulation methods

Methods	Examples	M&S	Service-orientation	System Engineering
F1	DUNIP, DEVS/SOA, SOAD, D-CD++	Unified DEVS model specification. DEVSML for platform independent models. SOAD can model & simulate service-based software & hardware systems. DEVS simulation protocol.	Simulators as services, models as resources in DUNIP. No coordinator services. Session management, configuration, simulation modeling & control, and retrieving data service interfaces in D-CD++.	DUNIP has bifurcated model-continuity systems engineering methodology.
F2	DDSOS	PSML-S can model SOA systems. RTI as runtime infrastructure. Optimistic time synchronization.	Systems/environment simulation agent services & RTI services. Support dynamic rebinding, re-composition, and re-architecture.	MDA and service-oriented systems engineering (SOSE) support.
F3	Service oriented HLA, HLA Evolved Web Service API etc.	BOM & modular FOM facilitate interoperability levels of models. Low bandwidth, uncertainty & dynamic properties need considering.	Web-Enabling HLA for communication, HLA interface specification, federate interface & 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.
F4	XMSF and profiles	The M&S focus area of XMSF.	The Web/XML, Internet/Networking focus area of XMSF.	N/A
F5	Cosim-Grid, SOAr-DSGrid, G-HLAM, SOHR	Simulation components, HLA/RTI services, computing & storage resources can be Grid services.	Focus on management of distributed computing resources. Based on Grid middleware.	Not clear
F6	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 & databases as services. Ontology method focuses on service UDDI, composition & fault-tolerant.	Have some development and usage procedures.

F1=Formalism based, F2=Model driven, F3=Interoperability protocol based, F4=XMSF, F5=OGSA based, F6=Other approaches.

Table 2 Advantages and Limitations of classical service-oriented simulation methods

Methods	Advantages	Limitations
F1	Mature formalism with long history. Strong presentation capability to various systems. Rigorous theoretical basis and mathematical semantics.	Too abstract & hard to follow by users. Have not been widely recognized by industrial & academic standards. Primarily for educational use. Simplicity, convenience & performance need to be improved.
F2	Model-driven, excellent dynamic composability and SOSE support.	Focus on service oriented software development. Limited simulation capabilities. Theory, efficiency & applications to be improved.
F3	Worldwide recognized IEEE standards. Solid research and practice foundations. HLA Evolved new standards	Revision while not the revolution of HLA may constrain further development. Lower levels of interoperability. Conflicts between SOA & HLA in service granularity.
F4	Pioneer technical frame; separate focus areas, issues & techniques.	Lack of concrete standards, products & systems engineering support. Has been terminated.
F5	Resource dynamic allocation, load balancing & fault tolerance. Transparency.	Needs grid middleware. M&S theoretical basis, systems engineering, performance & full use of SOA to be improved.
F6	Integration & interoperation of heterogeneous applications in Si3. UDDI & semantic composition in ontology methods.	Few publications & not mature. Need further research in the M&S dimension especially VV&A, states & time management of simulation service.

F1=Formalism based, F2=Model driven, F3=Interoperability protocol based, F4=XMSF, F5=OGSA based, F6=Other approaches.

The three dimensions comprise a reference model for a service-oriented simulation (Figure 1). 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 [25] and are inspired by the ontology of each discipline. To reveal the upper ontology and epistemology, we inspect the crossover of the three dimensions aggressively from a 1D, 2D, and 3D perspectives. Our 3D reference model is also inspired by, but differs from, the methodology of Morphological Analysis [26] and 3D morphology of systems engineering [27]. 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.

3.2 One-Dimensional Ontological Implication

Ontologies are often captured by a set of concepts and their relations. A 1D view enables us to look at each fundamental sub-ontology/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.

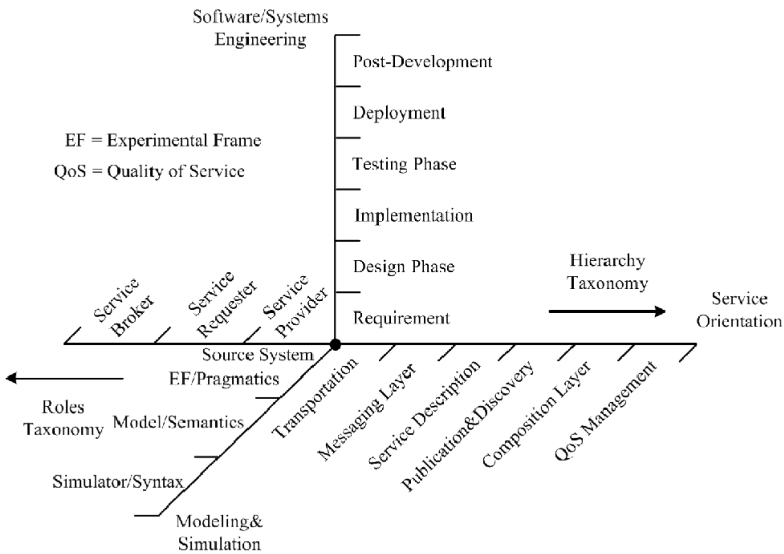

Fig. 1 A reference model for service-oriented simulation

3.2.1 M&S Dimension/Ontology

Besides the source system, the basic concepts in M&S [28] 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 concepts.

Additionally, other views in M&S, in particular composability and interoperability are necessary complementary. The challenges and contributions 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. The syntactic and semantic composability [29,30], the LCIM [31-33], the layers of M&S [23], and the interoperability challenges of model-based information systems (e.g., complex military simulation systems) [34] can also be mapped to these three levels with some reformulation or different interpretations.

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.

3.2.2 Service-Orientation Dimension/Ontology

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 [35]. 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. The SOA [35] is a conceptual framework for the design of business enterprise systems, while Web services is the prevailing technology for implementing a SOA. Previous work [35] 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 quality of service (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 requirements such as performance, reliability, scalability, interoperability, and security. We sequence the elements by their decreasing importance on the scale in Figure 1.

3.2.3 Software/Systems Engineering Dimension/Ontology

Simulation systems usually include software, at least in part [36]. The "Simulation as software engineering" mode of simulation practice [37] 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. [36] show 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 complement 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]. 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 are reported in [39,40]. Note that design and implementation often receive preferential treatment in general research and practice.

3.3 Two-Dimensional Epistemological Implication

Epistemologies study how we come to know, define, represent, and convey knowledge. In contrast with the ontological properties of the 1D view, a 2D view has an epistemological nature. It inspects the domains consisting of the Cartesian product of two sub-ontologies/dimensions to reveal the known or unknown knowledge in the cross-discipline landscape.

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

(1) core issues/knowledge (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/knowledge (S), the important characteristics of service-oriented simulation; if they are missing, the framework will be heavily affected; and

(3) nice-to-have issues/knowledge (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 3, 4, and 5. The 2D tables can be used for a cross-consistency assessment [26] process. They identify the logical and empirical meaning of each cell that consists of a pair of elements from the compared dimensions.

Table 3 Narrow service-oriented simulation (M&S vs. Services)

	S1	S2	S3	S4	S5	S6	S7	S8	S9
MS1	N	N	N	N	N	N	N	N	N
MS2	S	C	C	C	C	C	S	S	N
MS3	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. S1=Broker, S2=Requester, S3=Provider, S4=Transport, S5=Messaging, S6=Description, S7=Publish&Discovery, S8=Composition, S9=QoS, MS1=EF/Pragmatics, MS2=Model/Semantics, MS3=Simulator/Syntax.

Table 4 M&S engineering (M&S vs. Engineering)

	SE1	SE2	SE3	SE4	SE5	SE6
MS1	N	N	N	N	N	N
MS2	S	C	C	S	S	N
MS3	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. SE1=Requirements, SE2=Design, SE3=Implementation, SE4=Testing, SE5=Deployment, SE6=Post-development, MS1=EF/Pragmatics, MS2=Model/Semantics, MS3=Simulator/Syntax.

Table 5 Service-oriented engineering (Services vs. Engineering)

	S1	S2	S3	S4	S5	S6	S7	S8	S9
SE1	S	S	S	S	S	S	S	S	N
SE2	S	C	C	C	C	C	S	S	N
SE3	S	C	C	C	C	C	S	S	N
SE4	S	S	S	S	S	S	S	S	N
SE5	S	S	S	S	S	S	S	S	N
SE6	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. S1=Broker, S2=Requester, S3=Provider, S4=Transport, S5=Messaging, S6=Description, S7=Publish&Discovery, SE1=Requirements, SE2=Design, SE3= Implementation, SE4=Testing, SE5=Deployment, SE6=Post-development.

3.3.1 Narrow Service-Oriented Simulation

The Cartesian product of the M&S and service-orientation onlotogies/dimensions let us come to know service-oriented simulation in a narrow sense (Table 3). 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 [22] belong to this category. This 2D space has two epistemological 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 [26].

Capturing M&S and SOAs as discrete ontological elements and crossing them, produces some interesting epistemological observations. (i) From a 1D view, the headings of the first column in Table 3 identify the discrete ontological 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 [41]. A SOA also has other detailed taxonomies. The crossover of further discrete elements with their new relations can facilitate further research on the reusability and composability of M&S services.

3.3.2 M&S Engineering

The Cartesian product of the M&S and software/systems ontologies/dimensions provides an M&S engineering epistemology (Table 4) that applies engineering principles to traditional M&S as in, for example, the classical IEEE HLA Federation Development and Execution Process and VV&A standards. 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 [23].

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.

3.3.3 Service-Oriented Engineering

The Cartesian product of service-orientation and software/systems ontologies/ dimensions creates a service-oriented engineering epistemology (Table 5). 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 engineering [42] and service-oriented software engineering [43]. In particular, these authors discussed the impact of the SOA paradigm on classical software/systems engineering principles and practices.

3.4 Three-Dimensional Epistemological Implication

Despite the partial ontological and epistemological perspectives on the 1D and 2D interpretation, the 3D view illustrated in Figure 2 provides a complete multi-perspective epistemology of a service-oriented simulation. The whole 3D space consists of the Cartesian product of all three ontologies/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/knowledge in service-oriented simulation. This cube represents ‘service-oriented M&S engineering’, also called ‘general 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, the philosophical foundations of service-oriented simulation must cover the whole 3D space demanded by the 3D model.

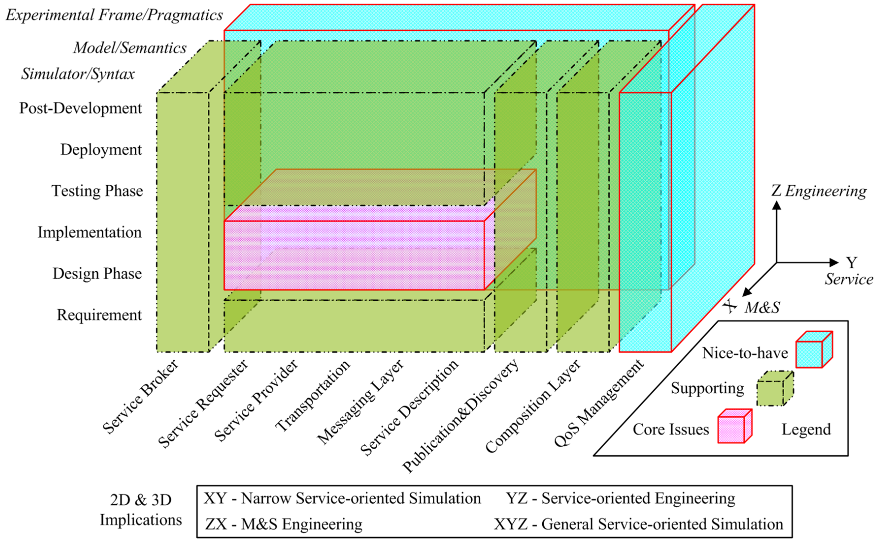

Fig. 2 Three-dimensional reference model for service-oriented simulation

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 knowledge or activities. Some sub-phases or steps can be added by multiple discipline experts using Cartesian products so that potential new knowledge or research issues can be discovered. Examples of possible new knowledge/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.

3.5 Descriptive and Prescriptive Roles

Engineering methods and their ontology, epistemology, and teleology distinguish characterization (description) and mandatory (prescription) [11, 44]. 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 two roles can be combined to show the potential and possible future directions of classical frameworks. The first role shows what cells have been covered in the 3D space, while the other shows what 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.

4 Teleological Perspectives on the Unifying Framework

Teleology is the study of purpose and purpose-driven actions that result in methods. Teleology has both the referential and methodological characteristics. In context of this chapter, the former focuses on real world systems to be simulated, while the latter emphasizes on modeling and simulation techniques. Based on the review and the 3D reference model, this section shows the teleology-driven activities/practice of the unifying methodology to describe, compare, and prescribe various frameworks.

4.1 Teleology Driven Recommended Practice

The unifying framework can be built from existing approaches by merging the similar, equivalent, or complementary capabilities they provided in all three dimensions. The teleology-driven 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 beyond the scope of this chapter.

(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. Section 4.2, 4.3, and 4.4.1 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 4.4.2.

(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 4.4.2 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 4.4.2.

The teleology and teleology-driven activities are revealed in the lifecycle of the recommended practice above. It facilitates to identify what do we need (purpose and 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/teleology. We apply the steps to the reviewed frameworks in the following subsections. Note that we focus only on the methodological teleology 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 specific referential teleology of the ad-hoc frameworks [3,4].

4.2 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 of our previous work [25]. In this chapter, we only show the primary 3D views in the Appendix Tables A1-A3. 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-A3) 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.

4.3 Gaps of Existing Frameworks

Using the prescriptive role of the 3D model as well as the analysis 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.

4.4 Teleology Driven Selection of Frameworks and Gaps Filling

4.4.1 Frameworks Selection

Based on the teleological 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 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. Note that the ontology and Si3 frameworks are not as mature as the others thus far. In sum, the formalism-based approach is the most mature one from an M&S theory perspective, while the interoperability protocol based method has the most potential in the practice.

4.4.2 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 1D, 2D, and 3D views. 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, the EF/Pragmatics is 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 are reported by Tolk et al. [45].

(3) Remaining gaps for future research

There are still some gaps (unknown knowledge or practice) 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 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.

5 Conclusions and Future Work

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.

In this chapter, we investigate service-oriented simulation frameworks from the ontological, epistemological, and teleological perspectives. We propose a unifying framework derived from the review of specific frameworks. With a referential (for real world) nature, the reviewed particular ontology, epistemology, and teleology of specific frameworks lead to various formalisms, techniques, and implementations. With a methodological (for abstract methods) nature, the ontology, epistemology, and teleology of the unifying framework build a systematical philosophical foundation for the service-oriented simulation paradigm. The unifying framework is first applied to two specific approaches [46]. Afterwards it is extended and further applied to some other prevailing methods [25]. It shows the unifying framework can, in turn, facilitate the classification, evaluation, selection, description, and prescription of the known or proposed frameworks.

Besides our former recommendations for the service-oriented simulation paradigm [25], there is also much future work from the context of the chapter. The referential and methodological nature, as well as the description and prescription roles of the ontology, epistemology, and teleology perspectives of the frameworks can be further investigated on more real world systems and abstract methods. Despite some specific frameworks, such as the DUNIP and DDSOS, support net-centric intelligent M&S applications, a better understanding of ontology, epistemology, and teleology of the frameworks are also necessary.

Acknowledgments. The authors thank Prof. Dr. Andreas Tolk for the invitation of the book chapter contribution. This work was partly supported by the National Natural Science Foundation of China [grant numbers 61074107, 60974073, 60974074, 91024015, 71031007]. The chapter is based on the author's paper "Wang WG, Wang WP, Zhu YF, Li Q (2011) Service-oriented simulation framework: An overview and unifying methodology. SIMULATION 87 (3):221-253." copyright © 2011 by the authors, the Society for Modelling and Simulation International, and Sage Publications Ltd. Reprinted by Permission of SAGE. We rewrite the chapter from the ontological, epistemological, and teleological perspectives.

References

1. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, United States (2005)
2. Chen, Y.: Modeling and Simulation for and in Service-Orientated Computing Paradigm. Simulation 83(1), 3–6 (2007), doi:10.1177/0037549707079218
3. Mittal, S.: DEVS Unified Process for Integrated Development and Testing of Service Oriented Architectures. Ph.D Dissertation, The University of Arizona, United States – Arizona (2007)
4. Fan, C.: DDSOS: A Dynamic Distributed Service-Oriented Modeling and Simulation Framework. Ph.D Dissertation, Arizona State University, United States – Arizona (2006)
5. Wang, W.G., Yu, W.G., Li, Q., Wang, W.P., Liu, X.C.: Service-Oriented High Level Architecture. In: European Simulation Interoperability Workshop, Edinburgh, Scotland. Simulation Interoperability Standards Organization (2008)
6. Li, B.H., Chai, X.D., Di, Y.Q., Yu, H.Y., Du, Z.H., Peng, X.Y.: Research on Service Oriented Simulation Grid. In: Proceedings of Autonomous Decentralized Systems, ISADS 2005, Chengdu, China, pp. 7–14 (2005), doi:10.1109/ISADS.2005.1452008
7. Zhang, T.: Research on Key Technologies of Service-Oriented Semantically Composable Simulation. Ph.D Dissertation, Changsha: National University of Defense Technology, China (2008)
8. Naval Postgraduate School MOVES Institute: eXtensible Modeling and Simulation Framework (XMSF) (2004), `http://www.movesinstitute.org/xmsf`
9. Tolk, A.: M&S Body of Knowledge: Progress Report and Look Ahead. SCS Magazine 2(4) (2010)
10. Turnitsa, C., Padilla, J.J., Tolk, A.: Ontology for Modeling and Simulation. In: Winter Simulation Conference, pp. 643–651 (2010)
11. Hofmann, M., Palii, J., Mihelcic, G.: Epistemic and Normative Aspects of Ontologies in Modelling and Simulation. Journal of Simulation 5(3), 135–146 (2011)
12. Sarjoughian, H., Kim, S., Ramaswamy, M., Yau, S.A.: Simulation Framework for Service-Oriented Computing Systems. In: Winter Simulation Conference, pp. 845–853 (2008)
13. Wainer, G.A., Madhoun, R., Al-Zoubi, K.: Distributed Simulation of DEVS and Cell-DEVS Models in CD++ Using Web-Services. Simul. Model. Pract. Theory 16(9), 1266–1292 (2008), doi:10.1016/j.simpat.2008.06.012
14. Tsai, W.T., Fan, C., Chen, Y., Paul, R.: A Service-Oriented Modeling and Simulation Framework for Rapid Development of Distributed Applications. Simul. Model. Pract. Theory 14(6), 725–739 (2006), doi:10.1016/j.simpat.2005.10.005

15. Möller, B., Löf, S.: Mixing Service Oriented and High Level Architectures in Support of the GIG. In: Spring Simulation Interoperability Workshop, San Diego, California. Simulation Interoperability Standards Organization (2005)
16. Wang, W.G., Xu, Y.P., Chen, X., Li, Q., Wang, W.P.: High Level Architecture Evolved Modular Federation Object Model. Journal of Systems Engineering and Electronics 20(3), 625–635 (2009)
17. Wang, W.G.: Service-Oriented Composable Simulation: Theory and Application for HLA Evolved. Ph.D Dissertation, Changsha: National University of Defense Technology, China (2011)
18. Brutzman, D., Zyda, M., Pullen, M., Morse, K.L.: 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)
19. Chen, X., Cai, W., Turner, S.J., Wang, Y.: SOAr-DSGrid: Service-Oriented Architecture for Distributed Simulation on the Grid. In: Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation, PADS 2006, pp. 65–73 (2006)
20. Rycerz, K., Bubak, M., Malawski, M., Sloot, P.: A Framework for HLA-Based Interactive Simulations on the Grid. Simulation 81(1), 67–76 (2005)
21. Pan, K., Turner, S.J., Cai, W., Li, Z.A.: Service Oriented HLA RTI on the Grid. In: 2007 IEEE International Conference on Web Services, ICWS 2007, pp. 984–992 (2007)
22. Strelich, T.P., Adams, D.P., Sloan, W.W.: Simulation-Based Transformation with the Service Integration/Interoperation Infrastructure. Technology Review Journal 13(2), 99–115 (2005)
23. Zeigler, B.P., Hammonds, P.E.: Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange. Academic Press, New York (2007)
24. Yilmaz, L.: A Strategy for Improving Dynamic Composability: Ontology-Driven Introspective Agent Architectures. Journal of Systemics, Cybernetics, and Informatics 5(5), 1–9 (2007)
25. Wang, W.G., Wang, W.P., Zhu, Y.F., Li, Q.: Service-Oriented Simulation Framework: An Overview and Unifying Methodology. Simulation 87(3), 221–253 (2011), doi:10.1177/0037549710391838
26. Ritchey, T.: Problem Structuring Using Computer-Aided Morphological Analysis. The Journal of the Operational Research Society 57(7), 792–801 (2006)
27. Hall, A.D.: Three-Dimensional Morphology of Systems Engineering. IEEE Transactions on Systems Science and Cybernetics 5(2), 156–160 (1969), doi:10.1109/TSSC.1969.300208
28. Zeigler, B.P., Praehofer, H., Kim, T.G.: Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, 2nd edn. Academic Press, USA (2000)
29. Petty, M.D., Weisel, E.W.: Composability Lexicon. In: Spring Simulation Interoperability Workshop. Simulation Interoperability Standards Organization (2003)
30. Szabo, C., Teo, Y.M., See, S.A.: Time-Based Formalism for the Validation of Semantic Composability. In: Winter Simulation Conference, pp. 1411–1422 (2009)
31. Tolk, A., Muguira, J.A.: The Levels of Conceptual Interoperability Model. In: Fall Simulation Interoperability Workshop, Orlando, Florida. Simulation Interoperability Standards Organization (2003)

32. Tolk, A., Diallo, S.Y., Turnitsa, C.D.: Applying the Levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and Composability for System-of-Systems Engineering. Systemics, Cybernetics, and Informatics 5(5), 65–74 (2008)
33. Tolk, A., Bair, L.J., Diallo, S.Y.: Supporting Network Enabled Capability by Extending the Levels of Conceptual Interoperability Model to an Interoperability Maturity Model. JDMS (2012), doi:10.1177/1548512911428457
34. Hofmann, M.A.: Challenges of Model Interoperation in Military Simulations. Simulation 80(12), 659–667 (2004)
35. Papazoglou, M.P., van den Heuvel, W.J.: Service Oriented Architectures: Approaches, Technologies and Research Issues. The International Journal on Very Large Data Bases 16(3), 389–415 (2007), doi:10.1007/s00778-007-0044-3
36. McKenzie, F.D., Petty, M.D., Xu, Q.: Usefulness of Software Architecture Description Languages for Modeling and Analysis of Federates and Federation Architectures. Simulation 80(11), 559–576 (2004), doi:10.1177/0037549704050185
37. Robinson, S.: Modes of Simulation Practice: Approaches to Business and Military Simulation. Simul. Model. Pract. Theory 10(8), 513–523 (2002)
38. IEEE (2008) Systems and Software Engineering - System Life Cycle Processes. 15288-2008
39. Mei, H., Shen, J.R.: Progress of Research on Software Architecture. Journal of Software 17(6), 1257–1275 (2006)
40. Jamshidi, M.: System of Systems Engineering—Innovations for the 21st Century. John Wiley & Sons, New York (2009)
41. Yilmaz, L.: On the Need for Contextualized Introspective Models to Improve Reuse and Composability of Defense Simulations. JDMS 1(3), 141–151 (2004)
42. Tsai, W.T.: Service-Oriented System Engineering: A New Paradigm. In: Proceedings of the 2005 IEEE International Workshop on Service-Oriented System Engineering, SOSE 2005, Beijing, China, pp. 3–6 (2005), doi:10.1109/SOSE.2005.34
43. Tsai, W.T., Bai, X.Y., Chen, Y.N.: On Service-Oriented Software Engineering. Tsinghua University Press, Beijing (2008)
44. Wang, W.G., Tolk, A., Wang, W.P.: The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S. In: SCS Spring Simulation Multiconference, SpringSim 2009, San Diego, CA, USA, pp. 375–384. ACM (2009)
45. Tolk, A., Diallo, S.Y., King, R.D., Turnitsa, C.D.: A Layered Approach to Composition and Interoperation in Complex Systems. SCI, vol. 168, pp. 41–74. Springer (2009)
46. Wang, W.G., Wang, W.P., Zander, J., Zhu, Y.F.: Three-Dimensional Conceptual Model for Service-Oriented Simulation. Journal of Zhejiang University Science A 10(8), 1075–1081 (2009)

Appendix Tables

Table A1 Comparison of frameworks from model/semantics's perspective (3D view)

	S1	S2	S3	S4	S5	S6	S7	S8	S9
SE1			F1	$F2^T$	F1	F1, M1			
SE2	$F2^T$, O2	F1 (User), $F2^T$, M1(PSML^T)	F1, $F2^T$, M1(PSML^T), G1, O1	F1, $F2^T$, M1(PSML^T)	F1, $F2^T$, M1(PSML^T)	F1(DEVSML), $F2^T$, M1(PSML^T), O1, O2	$F2^T$, O2	F1(SES, static), $F2^T$ (static), M1(PSML^T), O2	$F2^T$, O2
SE3	$F2^T$, O2	F1 (User), $F2^T$, M1(PSML^T)	F1, $F2^T$, M1(PSML^T), G1, O1	F1, $F2^T$, M1(PSML^T)	F1, $F2^T$, M1(PSML^T)	F1(DEVSML), $F2^T$, M1(PSML^T), O1, O2	$F2^T$, O2	F1(SES, static), $F2^T$ (static), M1(PSML^T), O2	$F2^T$, O2
SE4			F1, M1			F1			M1
SE5			F1, M1	F1, M1	F1, M1	F1, M1		M1	M1
SE6			M1	M1	M1	M1		M1	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. S1=Broker, S2=Requester, S3=Provider, S4= Transport, S5=Messaging, S6=Description, S7=Publish&Discovery, S8=Composition, S9=QoS, SE1= Requirements, SE2=Design, SE3=Implementation, SE4=Testing, SE5=Deployment, SE6=Post-development, F1= DUNIP, F2=SOAD, M1=DDSOS, G1=OGSA based frameworks, O1=Si3, O2=Ontology driven frameworks, SES=System Entity Structure, DEVSML=DEVS Modeling Language.

Table A2 Comparison of frameworks from simulator/syntax's perspective (3D view)

	S1	S2	S3	S4	S5	S6	S7	S8	S9
SE1					F1				
SE2	M1, O2	F1 (User), M1	F1, F3, M1, X1, I1, G1, O1	F1, M1, X1, I1, G1	F1, F3, M1, X1, I1, G1	F1, F3, M1, X1, I1, G1, O1, O2	M1, O2	M1, O2	M1, O2
SE3	M1, O2	F1 (User), M1	F1, F3, M1, X1, I1, G1, O1	F1, M1, X1, I1, G1	F1, F3, M1, X1, I1, G1	F1, F3, M1, X1, I1, G1, O1, O2	M1, O2	M1, O2	F3, M1, O2
SE4			I1						
SE5			F1, M1	F1, M1	F1, M1	F1, M1		M1	M1
SE6			M1	M1	M1	M1		M1	M1

The increasing gray intensity of the cells identifies nice-to-have, supporting, and core issues, respectively. S1=Broker, S2=Requester, S3=Provider, S4=Transport, S5=Messaging, S6= Description, S7=Publish&Discovery, S8=Composition, S9=QoS, SE1=Requirements, SE2=Design, SE3=Implementation, SE4=Testing, SE5=Deployment, SE6=Post-development, F1= DUNIP, F3=D-CD++, M1=DDSOS, I1=SOHLA, X1=XMSF, G1=OGSA based frameworks, O1=Si3, O2=Ontology driven frameworks, SES=System Entity Structure, DEVSML=DEVS Modeling Language.

Table A3 Comparison of frameworks from EF/pragmatics's perspective (3D view)

	S1	S2	S3	S4	S5	S6	S7	S8	S9
SE1						F1			
SE2	O2		M1	F1, M1	F1, M1	F1, M1	O2	M1	$F2^{T}$, F3, M1, G1, O2
SE3	O2		M1	F1, M1	F1, M1	F1, M1	O2	M1	$F2^{T}$, F3, M1, G1, O2
SE4									
SE5				F1	F1	F1			
SE6									

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. S1=Broker, S2=Requester, S3=Provider, S4=Transport, S5=Messaging, S6=Description, S7=Publish&Discovery, S8=Composition, S9=QoS, SE1=Requirements, SE2=Design, SE3=Implementation, SE4=Testing, SE5=Deployment, SE6=Post-development, F1= DUNIP, F2=SOAD, F3=D-CD++, M1=DDSOS, I1=SOHLA, X1=XMSF, G1=OGSA based frameworks, O2=Ontology driven frameworks.