JDMS

Journal of Defense Modeling and Simulation: Applications, Methodology, Technology
2016, Vol. 13(4) 367–380

DOI: 10.1177/1548512915621972
dms.sagepub.com
SAGE

Content analysis to classify and compare Live, Virtual, Constructive simulations and System of Systems

Saikou Y Diallo*, Jose J Padilla*, Yiannis Papelis*, Ross Gore, and Christopher J Lynch

Abstract
In this paper, we examine to what extent Live, Virtual, Constructive (LVC) simulations can be used to model and simulate a System of Systems (SoS) by (1) establishing LVC and SoS domain profiles based on concepts discussed in each community and (2) finding overlapping concepts. To do so, we apply content analysis to the Simulation Interoperability Standards Organization (SISO) conference papers to identify the driving concepts and relationships that are most associated with LVC systems. Similarly, we apply content analysis to the IEEE/Systems Man and Cybernetics Conference on System of Systems Engineering (SOSE). We use the Simulation Interoperability Standards Organization dataset to train a learner about LVC and automatically discover the LVC concepts and relationships within the SoS literature. Similarly, we use the SoS Engineering dataset to train a learner about SoS and use it to discover the concepts and relationships of SoS in the LVC literature. We show the two domains present some overlap on the technical side. However, LVC does not consider what the SoS literature considers crucial: the human/social component. This omission, or assumption, suggests that (1), although the SoS body of knowledge has some theoretical work, there is not enough work that transitions theory to practice and (2) LVC does not implement full SoS concepts but reflects traditional system approaches.

Keywords
content analysis, Live, Virtual, Constructive (LVC) simulation, System of Systems (SoS), systems engineering

1. Introduction

Challenges facing organizations adapting new technology can be divided into technical and organizational. Technical challenges are typically well-defined and specific and as a result are well studied and reasonably well understood within their respective fields. Organizational challenges on the other hand are inherently ill-defined and partially originate because of the interactions among several units such as human resources, management, engineering, etc. Systems engineering is used to study the interrelations and interdependencies between technical and organizational issues. Potential problems and pitfalls stemming from the interplay between organizational and technical issues are little understood, often ignored, or simply go unnoticed resulting in costly failures that are frequently and erroneously attributed to technology. Recently, there has been a push to combine the technical and organizational components to form what is known as a System of Systems (SoS).[1] SoS approaches provide alternatives to traditional approaches by considering a holistic view of the SoS, thereby giving system developers more opportunity to detect potential problems.[2]

SoS is defined as a ''large-scale concurrent and distributed system that [is] comprised of complex systems'' (Sahin et al., p. 1[3]). In other words, the individual systems within a SoS can operate on their own; however, the systems are networked together for the purpose of working towards a

Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA, USA

*SCS member

Corresponding author:
Saikou Y. Diallo, Virginia Modeling, Analysis and Simulation Center, 1030 University Blvd, Suffolk, VA, 23435, USA.
Email: Sdiallo@odu.edu

common goal.[4] As a group these systems provide an increased capability over the sum of the individual components,[5] hence the consideration of the holistic perspective. Some characteristics of SoS that have been identified in the literature and provide insight into the engineering challenge of SoS are:

- Systemic-Pluralist Context:[6,7] A SoS is open, purposeful, and consists of partially observable systems that have many involved decision makers.
- Operational and Managerial Independence:[8] A SoS has the characteristic that despite both types of organizations being controlled with a top-down approach, sub-systems function with a high level of independence. Each sub-system must fulfill a purpose and be managed for that purpose.
- Complexity:[9] A SoS has a large size in terms of sub-systems, sub-components, different types (technical, social), different functions, and non-linear relation among these systems/components, each pursuing their own goal while contributing to a common goal.
- Geographic Distance:[10] A SoS is usually geographically distributed.
- Evolutionary:[8,11] A SoS is never fully formed or complete as it changes through time in terms of structure and function.

These characteristics show how difficult it is to engineer or model a SoS even partially. For instance, how does one guarantee evolutionary characteristics? This suggests that SoS, although present in reality, are organically created by methods such as company acquisitions, mergers, and globalization among others. This presents a serious methodological challenge in studying a SoS especially when human/social considerations need to be taken into account.

One method for studying or engineering a SoS through simulation is the model-based approach where the individual systems within the SoS are captured as separate models that are then federated and executed in an orchestrated manner.[12] The use of models also facilitates the ability to display the structure of innovative solutions, provides a basis from which to evaluate ideas, and helps in identifying models that potentially can be reused to address new problems.[12,13] The model-based approach may rely on different modeling paradigms, such as agent-based modeling or system dynamics to capture the SoS at different levels depending on the characteristics of the system and the questions to be answered.[13] One instantiation of the model-based approach is to use the Discrete Event System Specification (DEVS) to formally describe each system in the SoS.[3] The resulting systems are coupled together through the use of the Extensible Markup Language (XML) to form the SoS. The orchestration of the SoS is natively supported through the mathematical representations of the systems and their couplings.[14] However, the model-based approach is technical in nature and assumes an objective identification of the problem, and corresponding solution, in reality via consensus or imposition.

An instantiation of the model-based approach is the use of Live, Virtual, Constructive (LVC) systems to model and simulate a SoS. LVC involves live people role playing with a live instructor (live), live people interacting with simulated entities (virtual), or simulated entities interacting with one another in a simulated environment (constructive). The use of LVC in the military domain has been the subset of a special issue in JDMS and the approach has been shown to be useful in testing and analysis,[15,16] training,[17] verification,[18] and modeling counter-insurgency.[19] The key assumption, when considering LVC as a viable form for designing and using SoS, is that the interactions between the live, virtual, and constructive components may account for the human/social aspect of SoS. We will show that this not reflected in the literature.

Although there are frameworks and approaches for developing an LVC system,[20] we do not have a clear understanding of the relationships between LVC and SoS in terms of what properties of the SoS should be reflected in the LVC for it to be a viable representation of a SoS. Notice we differentiate between LVC systems and LVC simulations. The former refers to the simulators, interoperability infrastructure, and simulation components needed to build LVC systems like the Live, Virtual, Constructive Architecture Framework (LVCAF) effort.[21] The latter refers to the use/application of these systems in exercises like the Ulchi-Freedom Guardian and the NATO pathfinder program.[22] In short, we need to better understand the relationship between LVC systems, LVC simulations, and SoS to gain a better understanding of the processes for constructing a SoS and how they can be supported by M&S.[23]

To identify what properties of SoS are reflected in LVC we use content analysis and machine learning on both the SOSE and LVC literatures. For SoS, we choose the IEEE/Systems Man and Cybernetics Conference on System of System Engineering (SOSE) conference because of its exclusive focus on SOSE. For LVC, we conduct a similar study on a subset of SISO that deals with LVC in order to understand what concepts and relationships underlie the LVC literature. We choose SISO because it has the most comprehensive body of knowledge in LVC practice. Content analysis is used to profile these two sets of papers by identifying salient concepts and their relationships. Machine learning is used to train a learner on the concepts of LVC and apply it to the SOSE corpus to identify the LVC concepts that are also SoS concepts. We use our findings to identify conceptual gaps between LVC and SoS and present areas that need to be explored in order to make LVC applicable to SoS.

2. Content analysis and machine learning

Content analysis is the process of making valid inferences from text (or textual forms of communication) about the contents of the text.[24,25] Content analysis produces the themes and concepts contained within the text in a reproducible manner by identifying the frequency of attributes (independent variables) within the text (i.e. a term or a piece of the text) and the relative co-occurrence of the attributes. We use Leximancer version 4[26] as the text analytic tool to conduct content analysis on two groups of literature which pertain separately to the categories (dependent variables) of SoS and LVC. Leximancer analyses text to find the presence of themes and concepts pertaining to a defined category (i.e., SoS or LVC). A theme is a cluster of concepts that appear together often and help to categorize the concepts obtained from the body of text. Overall, content analysis provides a method for breaking a body of text into a smaller set of manageable categories that can be more easily explored to gain insight into the content. Content analysis is applied to many domains[27–32] to learn from various bodies of knowledge and provides the ability to make determinations about a field, journal, application domain, or compare different bodies of knowledge.

Machine learning is used to determine the frequency and strength of each concept with respect to the defined categories of interest as the categories serve as the variables of interest when conducting content analysis. The relative frequency is measured by taking the conditional probability that a concept (C_O) is present given a particular category (C_A). The equation for relative frequency is

$$P(C_O|C_A) = \frac{P(C_O \cap C_A)}{P(C_A)} \tag{1}$$

The strength is measured by taking the conditional probability of the category given a particular attribute, such as a concept (C_O). Given C_O is present within a section of text, the strength is the probability that the section of text containing C_O comes from a particular category (C_A).[26] The equation for strength is

$$P(C_A|C_O) = \frac{P(C_A \cap C_O)}{P(C_O)} \tag{2}$$

The prominence of a concept within the text can then be calculated by dividing the joint probability of a concept within a category by the product of the marginal probabilities. The prominence score uses Bayesian statistics to combine the strength and relative frequency scores and provides a correlation between the category and the attribute.[26] The equation for prominence is

$$\text{Prominence} = \left(\frac{(A_c + C_c)}{B_c}\right) / \left(\frac{A_c}{B_c} * \frac{C_c}{B_c}\right) \tag{3}$$

where A_c is the number of occurrences of the term A within the text and C_c is the number of occurrences of the category C within the text. The term B_c represents the total number of context blocks identified within the text.

3. Data

As mentioned, we use two datasets consisting of (1) all SOSE conference papers from 2006 to 2014 for a total of 609 papers and (2) all SISO conference papers from 2000 to 2010 which resulted in a total of 2825 documents. A comprehensive list of characteristics, identified from the literature, that pertain to SoS is shown in Table 1. Although all of these characteristics are not necessarily present in every SoS, they have been presented in the literature as defining properties that are either desirable when engineering an SoS or observable when in the presence of an SoS. We will use this table as the basis for analyzing our data.

Within the LVC set, we search for the words ''LVC'' and ''Live, Virtual, Constructive''. We save each paper that matches either the ''LVC'' string or the ''Live, Virtual, Constructive'' string anywhere starting from the title to the conclusion. This resulted in a total of 37 papers that are broken down by year in Table 2, including the SOSE papers. It is important to note that LVC papers account for slightly over 1% of the total papers in the past 14 years in SISO. Nonetheless, it is important to mine the information within this corpus to see how it can be reused in support of SoS.

4. Results

We begin by analyzing the SOSE and the LVC literature to find the main themes in each domain. These themes are construct-like aggregations of related concepts. Then we study the overlapping concepts when possible.

4.1 SOS literature

We use the characteristics in Table 1 as concept tags to see how frequently and prominently they occur within the SoS corpus. Overall, 12 themes emerge for the content analysis with the *System* theme as the most relevant to the SOSE corpus (100%). This is not surprising if SoS is seen as a special case of a system. The *System* theme comprises the concepts of *system, requirement, capability, level, management, application, environment, change, components, technology*, and *structure*. However, it is interesting to note that only 42% of the corpus is related to SoS, fourth behind *data* (63%), *model* (83%) and *system* (100%). The *System* theme captures the systems engineering process of architecting, designing, and managing a SoS in order to provide a service or perform a function. The *System* theme

Table 1. Characteristics of a system of systems.

Characteristic	Definition	Motivation	Reference source
Autonomy	The quality of being independent and having behaviors. Independence is the ability of the systems to make independent choices and the behaviors allows the systems to pursue and fulfill their purposes.	The SoS has a higher purpose than any of its constituent systems, independently or additively; however, the pre-existing systems within a SoS are indispensable to an SoS.	33
Belonging	The constituent systems fall within the same boundary for inclusion within the SoS.	Pre-existing systems may be usable by a SoS but they may first need to be updated to serve in a SoS.	33
Compatibility	The ability of the supporting infrastructure to support a SoS.	The objectives sought in a system redesign must be compatible with the system and the SoS as inconsistencies between the redesign approach and the objectives will not yield the desired effectiveness.	11, 34–36
Connectivity	The ability of a system to link with other systems.	Pre-existing systems considered for an SoS are likely heterogeneous and they need to conform to the connectivity protocols of the SoS. A SoS is highly dependent upon the ability to maintain connectivity between systems during operation.	4, 33, 37
Diversity	Noticeable heterogeneity among the systems through distinct or unlike elements or qualities in a group or variations of social and cultural identities among people existing together in an operational setting. This is similar to the complexity property of Bizub et al.[20]	Pre-existing systems likely have not been purposed to work together prior to inclusion in the SoS; however, diversity is needed for the SoS to achieve its higher purposes through the combination of its systems.	9, 33
Emergence	The appearance of new properties in the course of development or evolution.	A boundary is indispensable to a system; emergence requires a well-defined boundary; an SoS has dynamic boundaries but always clearly defined; an SoS should be capable of developing an emergent culture with enhanced agility and adaptability.	7, 8, 33, 37
Evolutionary	A SoS is never fully formed or complete. Development of these systems is evolutionary over time and with structure, function and purpose added, removed, and modified as experience with the system increases.	A SoS has changing requirements over time as objectives are met or new objectives are identified.	8, 11
Geographic distribution	There can be a wide geographic dispersion of SoS systems. This can limit the ability of the systems to only exchanging information and not energy or mass.	Geographic location can limit the ability of systems to communicate.	8, 10
Managerial independence	The component systems not only can operate independently, they generally do operate independently to achieve an intended purpose. The component systems are generally individually acquired and integrated and they maintain a continuing operational existence that is independent of the system of systems.	If a system within a SoS is removed from the SoS, then it continues to operate and tries to meet its established goals without the use of an overarching authority to guide the individual processes.	8, 23
Operational independence	A system of systems is composed of systems that are independent and useful in their own right. If a system of systems is disassembled into the component systems, these component systems are capable of independently performing useful operations independently of one another.	If a system within a SoS is removed from the SoS, then the system needs the ability to operate on its own to meet its established goals. The rules of the system guide its execution.	4, 8, 10

Column 1 provides the characteristics, Column 2 provides a definition of each characteristic, and Column 3 provides the motivations behind the characteristic within SoS. Column 4 lists the sources providing the information in the first three columns taken primarily from Sauser and Boardman[33] and Cherns.[34]

Table 2. The number of papers comprising the SoS corpus and LVC corpus.

Year	SoS corpus	LVC corpus
2000	N/A	0
2001	N/A	0
2002	N/A	0
2003	N/A	1
2004	N/A	3
2005	N/A	13
2006	58	6
2007	119	5
2008	79	9
2009	39	0
2010	61	0
2011	56	N/A
2012	99	N/A
2013	46	N/A
2014	52	N/A
Total	609	37

Column 1 displays the year, Column 2 displays the number of papers found within the Simulation Interoperability Standards Organization that pertain to LVC for that year, and Column 3 displays the number of papers comprising the SoS dataset from the IEEE/SMC Conference on System of Systems Engineering for that year. For any year that we do not have data for a specific dataset, the cell shows an "N/A".

intersects with the *Model* theme to indicate that modeling is an approach for designing, developing, and studying the processes involved in a SoS. The *Model* theme is related to the *Results* theme through the concept of *Simulation* which is an indication of the use of simulation to investigate use cases or otherwise optimize the quantitative aspects of the system and the SoS.

The *SoS* theme includes the concepts of *architecture, capability, engineering, development*, and *requirements* but also encompasses most of the characteristics of SoS shown in Table 1. This observation shows that the characteristics of SoS as obtained from a selected literature review are confirmed within the SoS corpus and that they are a good starting point to characterize SoS. However, it also shows that, although SoS are systems with special properties, the engineering of SoS follows the traditional systems engineering process. There is no evidence of a special systems engineering process for SoS which is a significant gap in light of all of the properties of SoS. We denote the gap as follows:

Gap1. *There is no evidence of a System of Systems Engineering process that is separate and distinct from the systems engineering process.*

The role of modeling and simulation is evident in the *model, system, results* triangle and the LVC approach to support SoS should be situated within that area. However, it is also apparent that the SoS concept is ambiguous and we need to expand it further.

We examine the concepts and relationships most prominent (using equation 3) within the concept of SoS from the literature and display the results in Table 3. In this case, these concepts are the unit of analysis, not themes. Table 3 shows that SoS is a term that can refer to SoS as an *architecture*; SoS as a *capability*; SoS *engineering management; complex* SoS, SoS *system engineering*, SoS *development*; SoS *requirements engineering*; SoS *approach*, SoS *modeling and simulation*, and SoS *process engineering*.

Table 3 generates further questions when considering related concepts and how they are used in SoS. For instance, let's explore the term evolutionary as related to development. This either suggests that the term evolutionary is loosely used or that we can develop evolutionary systems. It can be argued that a system might be designed with the potential to evolve. Whether or not it does is contingent on the system, its sub-systems, its relation to other systems, and their corresponding environments. It is noted that the concept of evolutionary is not related to capability. It is also noted that these concepts can be desired properties that SoS could have or might be observed in reality. Yet, these properties might be the result of organic growth and not the result of development efforts. A second interesting concept is that of emergence. Emergence in this case might be related to uncertainty, either epistemic or aleatoric,[38] instead of systemic behavior not captured by system components or sub-systems. This seems to be the case in the development-emergence relation as some behaviors are really unforeseen outputs, not necessarily emergent.

When we combine Tables 1 and 3, we observe that all properties of SoS are covered by at least one SoS concept as shown in Table 4.

However, the concept of SoS as a capability covers the largest number of properties which might be an indication that SoS *capability* is the most general concept of SoS. It is also interesting to note that the concepts of SoS *approach* and SoS *model* are not associated with any of the properties of a SoS. This represents a gap in the SoS body of knowledge since the SoS model is supposed to faithfully represent some aspects of the SoS and therefore should exhibit at least one of its properties.

Gap2. *There is no evidence that a System of Systems model has at least one property of a System of Systems.*

We also note that independence is a property that is almost always associated with SoS but is not commonly associated with a complex SoS and SoS engineering in this corpus. Table 4 leads us to observe that if LVC systems should be used to study SoS, we need to specify which concept of SoS is being modeled. Consequently, it is important to understand where LVC applies within the different understandings of SoS and more importantly, it is essential to determine whether the properties of SoS in

Table 3. The concepts identified within the SOSE body of knowledge.

Most prominent concepts	Related concept	Supporting text
Architecture	Independence	Independence has two different aspects[1]: operational independence and managerial independence. Operational independence imposes architectural constraints on the SoS.
Capability	Mission	Responsible for fielding the SoS) or chief architect that must make these system operate together to carry out a mission or provide a new capability? It is suggested that, in the future, the "operator" becomes the pseudo program manager or chief architect that must carry out the mission.
	Performance	The SoS paradigm presents a new school of thought in Systems Engineering. The driving force behind the desire to view these systems a SoS is to achieve higher capabilities and performance than would be possible with a traditional stand-alone system.
	Autonomy	The open-system paradigm allows evolution of CS capabilities to align increasingly with the goals of SoS, and the loose coupling ensures robustness and resilience of SoS as a whole, as well as protection for the openness of CSs. The looseness in interdependence can be achieved by independence of CSs, by such properties as autonomy and geographic distribution.
	Belonging	An additional characteristic of an unbounded SoS is the constituent systems choose to belong to the greater collective and are dependent on the value of the union. The capabilities of the participant and the expected capabilities and emergent behavior of the interoperability union in SoS define the characteristic of belonging.
	Compatibility	Measures and calculations of the abnormal state evolution based on compatibility test Firstly, to construct a State Vector of Cap (SoS) (SVC). Assuming the capabilities of SoS are divided according to certain state, vectors formed under the divisions are the state vectors of the capabilities of SoS.
	Connectivity	SoS connectivity is defined as "dynamically supplied by constituent systems with every possibility of myriad connections between constituent systems, possibly via a net-centric architecture, to enhance SoS capability".[5] The connectivity, platform diversity, and associated interoperability defines whether the SoS is bounded or unbounded.
	Diversity	However a SoS is much more because its parts, acting as autonomous systems, forming their own connections and rejoicing in their diversity, lead to enhanced emergence, something that fulfills capability demands that set a SoS apart. We summarize our thinking in Table 1 which also includes a set of cross references from our literature research where we believe others are articulating our chosen differentiating characteristics.
	Independence	Service orientation is becoming more common for SoS implementation as it supports operational independence, managerial independence, and geographic distribution of
Engineering	Emergent	There is consequently a need in model-based SoS engineering to specify constituent system behavior in a way that bounds the range of behaviors that can be relied upon without over constraining them, while still allowing the analyses needed to promote desirable and limit undesirable emergent behaviors. 1 www.compass-research.eu
Complex	Connectivity	In connectivity at any given level, and the heterogeneity of constituents, just to name a few. In particular, the complexity associated with level interaction in a SoS design context can quickly escalate to
	Distributed	Kotov[1] defines SoS as "large scale concurrent and distributed systems that are comprised of complex systems." Each individual system in Figure 1 is a complex system that works concurrently of the other systems.
	Emergent	In the Spectrum of System, Mitre put SoS between ordinary system (simple system) and complex system, meaning that SoS can either be simple system or complex system. However, if "emergent behavior" is applied to define SoS or complex system of systems, the position of SoS in the spectrum of system shall be within the complex system, as illustrated in Figure 1: complex System of Systems 1
	Evolutionary	A SoS framework can effectively guide this complex evolutionary
System	Autonomy	This autonomy is what separates the systems from the parts. Accordingly this definition of autonomy applies to the systems within the SoS as well as the SoS as a whole.
	Connectivity	Connectivity- The ability to stay connected to other constituent systems.[22] Boardman and Sauser[24] relate connectivity to interoperability amongst the legacy systems and possibly additions of new systems to the SoS.

(continued)

Table 3. (Continued)

Most prominent concepts	Related concept	Supporting text
	Independence	The ability of a system as part of SoS to make independent choices. This includes managerial and operational independence while accomplishing the purpose of SoS; Belonging.
Development	Emergent	The SoS approach allows for unforeseen emergent behavior which the system then eliminates by adaption. The SoSE approach can be considered a development of SE, increasing its effectiveness and allowing new challenges to be addressed.
	Evolutionary	(Not incremental development wherein their future behavior is known, and is just parceled out in increments.) How do we integrate these evolutionary systems that are using spiral development into a SoS?
	Independence	Referring to the issues raised by the independence and asynchronous development schedules of constituent systems, the SoS SEG states that SoS have addressed this conundrum in different ways. For example, "… a number of
Requirements		Therefore the second SoS requirement is consequent to evolving requirement. That is, the AUVs require modularity and reconfigurability
Approach		Alternatively, some scientists took a different approach by focusing on distinguishing characteristics rather than providing an abstract definition of the SoS. According to Boardman and Sauser,[12]'there are five."
Model		This is inferred from a systemigram of the USAF C2 Constellation which provides a profound overview of the SoS interoperability (SoSI) characteristics.[7] Interoperability models define three types of interoperability, shown in Figure 1, within a SoSI context programmatic, constructive, and operational.
Process		SoSE is executed (in conjunction with various SE processes) as coordinated by the SoSE Management process. The SoS is checked in the SoSE Test and Evaluation process.

Column 1 provides the main concepts identified from the SOSE body of knowledge. Column 2 shows the most frequently related concept and Column 3 shows supporting text from the corpus. Note, any citations listed within Column 3 reflect the text that Leximancer identifies to support the related concept within Column 2 and does not refer to our reference list, and some sentences may appear in an incomplete form due to how the supporting text is conveyed by Leximancer.

Table 4. SoS concepts and properties.

	Architecture	Capability	Engineering	Complex	System	Development	Requirements	Approach	Model	Process
Autonomy		X			X		N/A	N/A	N/A	N/A
Belonging		X					N/A	N/A	N/A	N/A
Compatibility		X			X		N/A	N/A	N/A	N/A
Connectivity	X	X		X			N/A	N/A	N/A	N/A
Diversity		X					N/A	N/A	N/A	N/A
Emergence			X	X		X	N/A	N/A	N/A	N/A
Evolutionary				X		X	N/A	N/A	N/A	N/A
Geographic distribution	X			X			N/A	N/A	N/A	N/A
Managerial independence	X	X			X	X	N/A	N/A	N/A	N/A
Operational independence	X	X			X	X	N/A	N/A	N/A	N/A

The columns indicate the concepts mostly associated with SoS through conceptual analysis and the rows are the characteristics of SoS from Table 1. Together they show that the concept of SoS as a capability covers the highest number of characteristics while the concept of SoS engineering only covers the characteristic of emergence. Cells marked "N/A" indicate that the content analysis did not find a concept to match with the SoS characteristic.

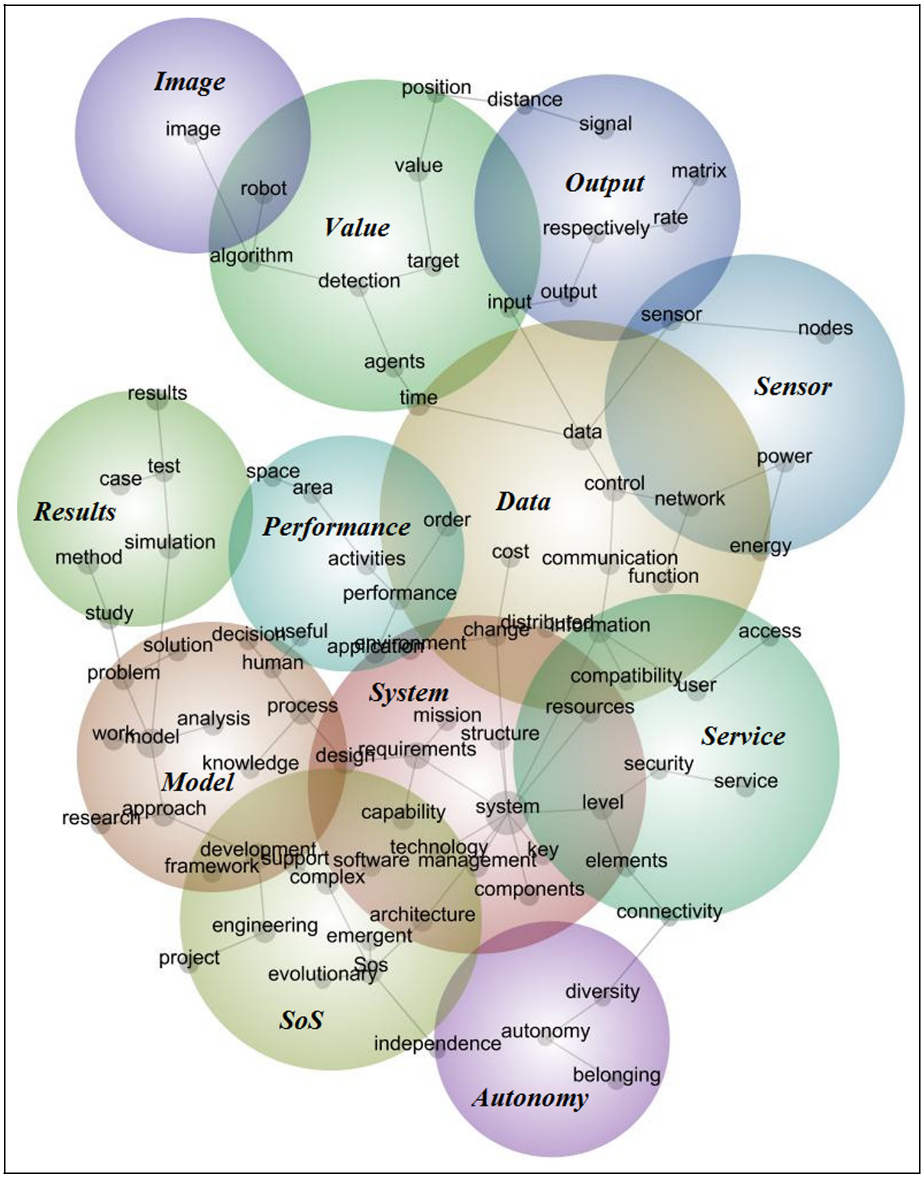

Figure 1. Concept map of the SOSE literature pertaining to SoS. The main themes within SOSE literature are identifiable by the circles and the theme name is in bold and italics within the circle. The remaining names represent the main concepts within each theme and the links connecting the concepts show the mapping of the concepts across themes. The color of the themes indicates the importance of the theme to the literature in a heat map fashion where the most important theme is in red, then orange, progressing down to blue. Concepts that appear in the overlap of more than one theme are highly connected to all of those themes. Color available online.

general are applicable to these different concepts of SoS. In the next section, we focus on the LVC corpus to examine its themes and concepts.

4.2 LVC literature

A content analysis of the LVC corpus reveals 12 themes which are *development, simulation, requirements, current, application, data, network, DIS, model, software, field*, and *JCATS*. The theme of *development* (shown in red in Figure 2) is the most important theme and the theme of *simulation* is the second most important theme. The focus of LVC is entirely based on implementation. The view of simulation deals with the application and development of the simulation and the requirements, data, and networking associated with the construction of the simulation. Supporting this claim are the themes of *DIS* (a standard for distributed computing), *JCATS* (a constructive simulation tool), and *software* appearing from the literature.

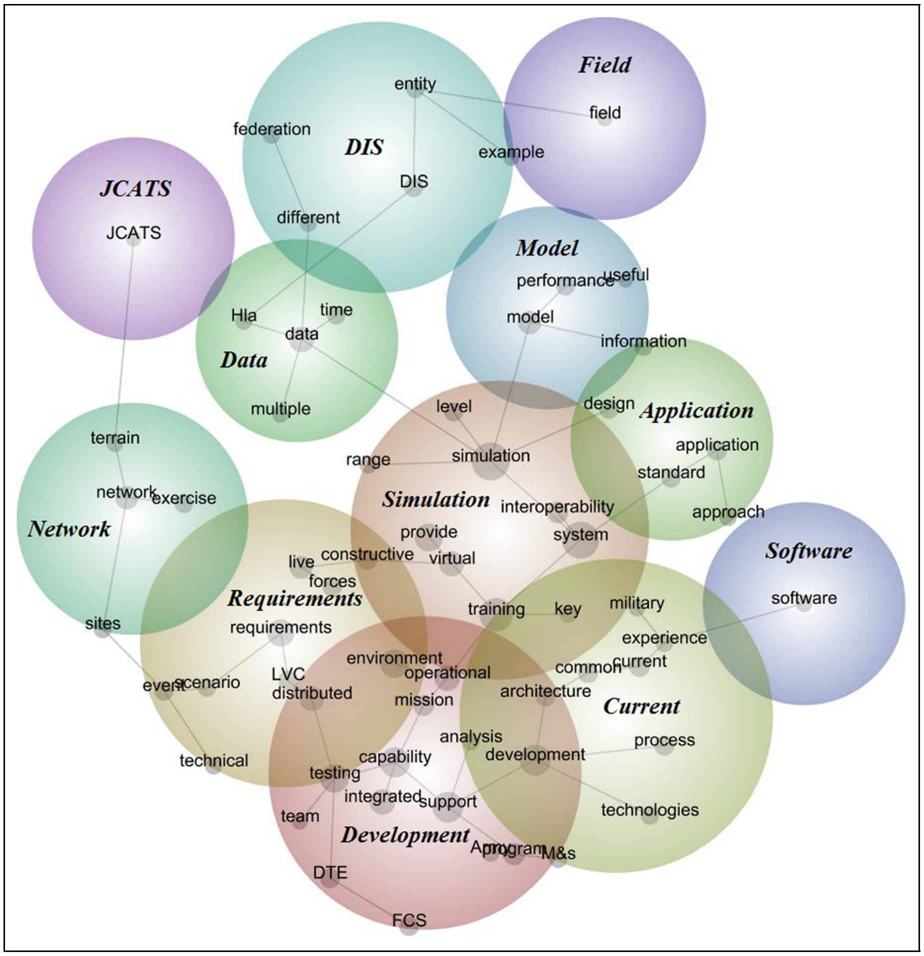

Figure 2. Concept map of the SISO literature pertaining to LVC. The main themes within the SISO LVC literature are identifiable by the circles and the theme name is in bold and italics within the circle. The remaining names represent the main concepts within each theme and the links connecting the concepts show the mapping of the concepts across the themes. The color of the themes indicates the importance of the theme to the literature in a heat map fashion where the most important theme is in red, then orange, progressing down to blue. Concepts that appear in the overlap of more than one theme are highly connected to all of those themes. LVC: Live, Virtual, Constructive; HLA: High Level Architecture; DTE: defense training environment; DIS: Distributed Interactive Simulation (a standard for distributed computing); JCATS: Joint Conflict and Tactical Simulation (a constructive simulation tool); FCS: future combat systems. Color available online.

The most prominent concepts within the LVC corpus include *integrated, distributed, environment, capability, testing, training*, and *requirements*. These concepts support that the focus of LVC is on the implementation of the LVC system in order to conduct a LVC simulation. We also note that *Live* appears within the text as its own concept but *virtual* and *constructive* are not themselves main concepts. Instead *virtual* and *constructive* appear as related concepts to other concepts within the text. However, the prominence of '*live* with *constructive*' and '*live* with *virtual*' is very high which leads us to conclude that live simulation is a key driver in the motivation to use LVC. The concept of *system* within the LVC corpus refers to LVC as a system. All of the main concepts along with their associated concepts obtained from the LVC corpus are provided in Table 5. These concepts are the characteristics of LVC that are required for implementing a model of a SoS.

The concept of *LVC* falls within the theme of requirements and the individual concepts of *live, virtual*, and *constructive* link the *requirements* theme directly to the *simulation* theme. The *requirements* of the LVC involve accounting for the scenario that will be used for training, the distributed components of the LVC, and the technical requirements of the system that allow for the necessary networking components of the LVC system. The LVC *data* are handled through a federation and involves the standards of HLA and DIS. The concept of

Table 5. The main concepts within the LVC subset of the SISO dataset.

Concept	Related Concept
Integrated	constructive distributed environment key live
Distributed	constructive environment live technical testing useful
Live	constructive environment terrain virtual
Environment	constructive scenario testing
Capability	constructive scenario
Testing	None
Training	None
Requirements	None
Simulation	None
System	None

Column 1 provides the main concepts identified from the LVC subset of the SISO dataset. Column 2 shows the concepts that are related to the main concept.

interoperability is connected to *simulation* through the themes of *requirements* and *development* and ultimately the concept of *system* and then *simulation*.

From Table 5, we observe that similarly to SoS, LVC is a term that can be referred to as LVC *capability*, LVC *environment*, LVC *system*, LVC *simulation*. We also note that for the most part LVC is used for training and testing and not for experimentation. This leads us to our third gap.

Gap3. *There is no evidence of the consistent use of LVC outside of training and testing.*

This gap implies that further research is needed in order to determine if and how LVC can be used to experiment with, optimize, or otherwise study SoS. A main difference between the SoS concepts and the LVC concepts is the prominence of *model* in the SoS corpus and *simulation* in LVC. The SoS corpus focuses on the conceptualization and design of a solution, whereas the LVC corpus focuses on the implementation and execution of a model. This identifies a key difference between the SoS literature and the LVC literature and leads us to our fourth gap.

Gap4. *There is no evidence of a Systems Engineering or a System of Systems Engineering process being employed when developing LVC systems.*

The concept of *model* within the LVC literature is much less prominent, whereas the concept of *simulation* is much more prominent compared to the SoS literature, indicating a shift in focus from design of the SoS to implementation of the LVC. This is an indication that LVC research can greatly benefit from theories and frameworks within the SoS body of knowledge and that SoS research can benefit greatly from lessons learned through LVC implementations.

Additionally, there are some overlapping concepts which appear in both corpora, including *capability, requirements*, and *system*. The SoS concept of *capability* refers to the capabilities associated with each of the systems within the SoS and the LVC concept of *capability* refers to the capabilities associated with constructive simulations and the scenario under which the LVC is utilized. The concepts of *system* differ in that SoS views a *system* as each individual system that comprises the SoS, whereas LVC views the collective LVC simulation as a *system*. The concept of *requirements* differs in that SoS requirements deal with architecture and the ability of the systems to function autonomously. For LVC the requirements are much more focused on the ability of the systems to communicate and ensure the continued ability of the LVC system to support an exercise. It is important to note that LVC is strongly related to *distributed* which is one of the key properties of SoS.

4.3 SOS and LVC overlap

So far, we know that SoS can refer to many concepts with different properties and we know that LVC systems are used as a capability for training and testing. In this section, we examine the concepts of LVC as they apply to SoS. We focus on the overlapping notion of capability and system and investigate whether a property of SoS is usually associated with it in the LVC corpus.

Table 6 shows that an LVC capability with the properties of autonomy, connectivity, evolution, geographic distribution, and independence can be used to replicate SoS capabilities. Similarly, an LVC system that has the properties of belonging, compatibility, and distribution can used to replicate SoS systems.

Table 7 presents a comparative view of the relationships between the main SoS concepts and the main SoS characteristics from the perspectives of both SoS and LVC. Several areas are identified where a concept fails to address any of the SoS characteristics from either the SoS or LVC perspective. In gray, we highlight the properties of SoS that are not found in LVC capabilities or systems.

Table 6. LVC properties by concept.

	Architecture	Capability	Engineering	Complex	System	Development	Requirements	Approach	Model	Process
Autonomy			N/A	N/A						
Belonging			N/A	N/A	X					
Compatibility			N/A	N/A	X	X	X	X	X	
Connectivity	X	X	N/A	N/A	X		X	X		X
Diversity			N/A	N/A	X		X			
Emergence			N/A	N/A						
Evolutionary		X	N/A	N/A		X		X	X	X
Geographic distribution	X	X	N/A	N/A	X	X		X	X	
Managerial independence		X	N/A	N/A		X			X	
Operational independence		X	N/A	N/A		X			X	

The main LVC concepts are displayed across the top row of the table and the SoS characteristics from Table 1 are displayed along the left column. The *X*s in the table show areas that overlap between the LVC and the SoS concepts. Columns that are marked N/A represent LVC concepts which do not have a consistent overlap with the SoS concepts.

Table 7. Side-by-side comparison between LVC and SoS concepts and characteristics.

	Architecture		Capability		Engineering		Complex		System		Development		Requirement		Approach		Model		Process	
	SoS	LVC	SoS	LVC	SoS	LVC	SoS	LVC	SoS	LVC	SoS	LVC	SoS	LVC	SoS	LVC	SoS	LVC	SoS	LVC
Autonomy			X						X											
Belonging			X							X		X								
Compatibility			X	X					X	X				X		X		X		
Connectivity		X	X				X			X				X		X				X
Diversity			X							X				X						
Emergence					X		X				X									
Evolutionary				X			X				X	X				X		X		X
Geographic Distribution		X		X			X			X		X				X		X		
Managerial Independence	X		X	X					X		X	X						X		
Operational Independence	X		X	X					X		X	X						X		

The main SoS concepts are displayed across the top row of the table and the SoS characteristics from Table 1 are displayed along the left column. Cells marked with an *X* indicate that the characteristic is often found in association with the corresponding concept for SoS or LVC.

Aside from geographic distribution, LVC capabilities and systems have starkly different properties and both lack diversity and emergence. We also note that LVC capabilities are evolutionary and distributed, whereas SoS capabilities are not usually linked with those concepts.

Since these concepts are identified as main concepts for SoS based on the SoS body of knowledge, these areas highlight major gaps for creating a SoS or for modeling a SoS using LVC. The main findings from this comparison are as follows:

- *Architecture*: There is no overlap between LVC architectures and SoS architecture. The LVC architecture frameworks focus solely on the technical aspects of implementing LVC systems. If these LVC systems are implementations of LVC models of SoS then an LVC system is a purposeful abstraction of SoS. Therefore, the LVC architecture should be aligned with the SoS architecture. This leads to us to the following gap.
  Gap5: *We need to study how to align LVC architectures with System of Systems architectures.* However, when it comes to SoS architectures, while mentioned in the body of knowledge, there is not one that abides by SoS properties and provides a template for starting a SoS from design to implementation and testing outside of systems engineering. Unlike LVC, there has not been a major SoS architectural effort.
- *Capability*: SoS and LVC capabilities overlap in terms of independence and compatibility. However, neither LVC nor SoS capabilities cover the property of emergence which leads us to our next gap.
  Gap6: *We need to study the property of emergence in SoS and LVC.* At the very least, we need to define what emergence is in the SoS and LVC contexts.
- *Engineering*: SoS engineering as we mentioned earlier does not appear to differ from systems engineering except for the fact that it is supposed to address emergence. Nonetheless, there is wide gap when it comes to LVC engineering in support of SoS. This leads us to our next gap.
  Gap7: *We need to study LVC engineering in support of SoS.* This means that the LVC and/or SoS community need to provide theories, methodologies, methods, and tools to make this possible.
- *Complex*: The notion of a complex SoS exists but the notion of a complex LVC does not. We know that the complexity in SoS, based on the analysis, refers to the environment in which the SoS is found and the other systems a SoS may be connected to considering evolutionary and geographically distributed properties. This leads us to our next gap.
  Gap8: *We need to study the complexity of LVC environments.* Findings in the study of complex SoS might be transferrable to complex LVC. Further, we need to systematically investigate the complexity of LVC environments in order to determine if they can be used in the study of complex SoS.
- *System*: LVC systems and SoS are well covered in their system aspects. However, there is a gap in the understanding of the emergence and evolutionary aspects in both cases and in the case of LVC the managerial and operational independence are ignored. This leads us to our next gap.
  Gap9: *We need to study the emergent and evolutionary properties of SoS and LVC systems.* This is important specially when considering the independence property in LVC systems.
- *Development*: LVC development covers mainly the technical and to some extent non-technical aspects, even though this is not reflected in LVC architectures (LVCAF) and frameworks (DSEEP). SoS development is focused on the independence, emergent, and evolutionary aspects of SoS. However, the emergent and evolutionary aspects are not reflected in SoS architectures. This emphasizes gaps 5 and 9.
- *Requirements*: SoS requirements focus on the functional, software, and design requirements of SoS. Similar to requirements engineering, we need SoS requirement engineering. This leads us to our next gap.
  Gap10: *We need to study requirements engineering for SoS and LVC.* The departing point is whether or not requirements engineering for a system is the same for a SoS.
- *Approach*: A SoS approach to dealing with real world situations is an important tenet of SoS. However, we do not associate any of the SoS properties with an SoS approach. This leads us to emphasize gaps 7 and 10 and to consider the following gap.
  Gap11: *We need to study the properties of a System of Systems approach.* This includes addressing questions such as do we need SoS if they are engineered as systems or can we even develop SoS approaches? Is it doable?
- *Model*: A SoS model does not satisfy any of the main SoS characteristics. A LVC model addresses the SoS characteristics of compatibility, evolutionary, distributed, and independence. The LVC simulation cannot serve as a reliable representation of the SoS unless it has the ability to represent the evolutionary aspects of the SoS systems. The LVC simulation must be able to function in a distributed

environment where the component systems remain compatible with the LVC even as they evolve and retain their ability to operate independent of the LVC. This leads us to our next gap.

Gap12: *We need to study System of Systems modeling.* Similar to gap 11, we still need to ask, can it be done? Do we need it?

- *Process*: A SoS process is focused on the results but we do not understand what properties to ascribe to them. For instance, how do the results change in a SoS with operational independence compared to the same SoS with a centralized authority? This leads us to the next gap.

  Gap13: *We need to study the properties of the results of applying System of Systems.* In other words, we need to establish the means to compare SoS under one or more SoS properties. We can always speculate that simulations provide great insight, but as mentioned, at least on the LVC side, much research is needed.

5. Conclusion

In this paper, we have shown that the term LVC and the term SoS can be decomposed into several concepts that need to be aligned. We have shown that a systematic application of content analysis can reveal several areas of research in the SoS and LVC domains. Finally, we have shown that, while LVC can be applied to SoS, there is still much left to learn in order for LVC to become a common method for studying SoS. In the future, we will track the SoS and LVC bodies of knowledge in order to explore whether they evolve in the same direction.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Hipel K, Jamshidi M, Tien J, et al. The future of systems, man, and cybernetics: application domains and research methods. *IEEE Trans Syst, Man, Cybern, Part C: Appl Rev* 2007; 37(5): 726-743.
2. Keating C, Padilla J and Adams K. System of systems engineering requirements: Challenges and guidelines. *Engng Mgmt J* 2008; 20(4): 24-31.
3. Sahin F, Jamshidi M and Sridhar P. A discrete event xml based simulation framework for system of systems architectures. In: *Proceedings of the 2007 IEEE International Conference on System of Systems Engineering (SoSE'07)*, 2007, pp.1-7.
4. Jamshidi M. Introduction to system of systems. In: Jamshidi M (ed.) *System of systems engineering – innovations for the 21st century*. Hoboken, NJ: Wiley, 2009, pp.1-20.
5. Gallant S and Gaughan C. Systems engineering for distributed live, virtual, and constructive (LVC) simulation. In: *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp.1501-1511.
6. Jackson M and Keys P. Towards a system of systems methodologies. *J Opl Res Soc* 1984; 35(6): 473-486.
7. Keating C. Emergence system of systems. In. Jamshidi M (ed.) *System of systems engineering – innovations for the 21st century*. Hoboken, NJ: Wiley, 2009, pp.169-190.
8. Maier MW. Architecting principles for systems-of-systems. *Syst Engng* 1998; 1(4): 267-284.
9. Bar-Yam Y. *Dynamics of complex systems*. Reading, MA: Addison-Wesley, 1997.
10. Eisner H, McMillan R, Marciniak J, et al. RCASSE: rapid computer-aided system of systems (S2) engineering. In: *Proceedings of the 3rd International Symposium of the National Council on Systems Engineering*, 1993, Vol. 1, pp.267-273.
11. Cherns A. Principles of sociotechnical design revisited. *Hum Relat* 1987; 40(3): 153-161.
12. Tolk A, Diallo S and Turnitsa C. Applying the levels of conceptual interoperability model in support of integratability, interoperability and composability for system-of-systems engineering. *J Systemics, Cybern Inf* 2007; 5(5): 65-74.
13. Eusgeld I, Nan C and Dietz S. 'System-of-systems' approach for interdependent critical infrastructures. *Reliab Eng Syst Saf* 2011; 96(6): 679-686.
14. Mittal S, Zeigler BP and Martin J. Implementation of formal standard for interoperability in M&S/systems of systems integration with DEVS/SOA. *Int Command Control C2 J, Special Issue: Modeling and Simulation in Support of Network-Centric Approaches and Capabilities* 2009; 3(1): 1-61.
15. Hodson DD and Hill RR. The art and science of live, virtual, and constructive simulation for test and analysis. *J Defense Model Simul: Appl, Method, Technol* 2014; 11(2): 77-89.
16. Mezzacappa E. Effectiveness testing of non-lethal weapons. *J Defense Model Simul: Appl, Method, Technol* 2013; 11(2): 91-101.
17. Dickerson C and Holden T. Relation oriented systems engineering framework for flight training. *J Defense Model Simul: Appl, Method, Technol* 2014; 11(2): 103-113.
18. Thoman D, Henley ER, O'Kula K, et al. An iterative process for verification of a meteorological data server. *J Defense Model Simul: Appl, Method, Technol* 2013; 10(4):411-423.
19. Monday P. Architecture of the counter insurgency experiment. *J Defense Model Simul: Appl, Method, Technol* 2009; 6(2): 69-77.
20. Bizub W, Wallace J, Ceranowicz A, et al. Next-generation live virtual constructive architecture framework (LVCAF). In: *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, Orlando, 2009.

21. Bizub W and Brandt J. Transitioning to the next generation (nextgen) defense training environment (DTE). In: *Proceedings of the 2011 Winter Simulation Conference*, 2011, pp.2564-2575.
22. Sokolowski JA and Banks CM (eds). *Handbook of real-world applications in modeling and simulation* (Vol. 2). Hoboken, NJ: John Wiley & Sons, 2012.
23. Tolk A and Rainey LB. Toward a research agenda for M&S support of system of systems engineering. In: Rainey L and Tolk A (eds) *Modeling and simulation support for system of systems engineering applications*. Hoboken, NJ: John Wiley & Sons, 2015, pp.583-592.
24. Berelson B. *Content analysis in communication research*. Glencoe, IL: The Free Press, 1952.
25. Krippendorff K. *Content analysis: An introduction to its methodology*. Thousand Oaks, CA: Sage, 2012.
26. Leximancer. Leximancer Manual: Version 4, Online resource document, https://www.leximancer.com/site-media/lm/science/Leximancer_Manual_Version_4_0.pdf. (2011, accessed 15 October 2014.
27. Chang YH, Chang CY and Tseng YH. Trends of science education research: An automatic content analysis. *J Sci Edu Technol* 2010; 19(4): 315-331.
28. Tsai CC and Lydia Wen M. Research and trends in science education from 1998 to 2002: A content analysis of publication in selected journals. *Int J Sci Edu* 2001; 27(1): 3-14.
29. Lee MH, Wu YT and Tsai CC. Research trends in science education from 2003 to 2007: A content analysis of publications in selected journals. *Int J Sci Edu* 2009; 31(15): 1999-2020.
30. Mustafee N, Katsaliaki K and Fishwick P. Exploring the modelling and simulation knowledge base through journal co-citation analysis. *Scientometrics* 2014; 98(3): 2145-2159.
31. Crofts K and Bisman J. Interrogating accountability: An illustration of the use of Leximancer software for qualitative data analysis. *Qual Res Acc Mgmt* 2010; 7(2): 180-207.
32. Noltemeyer AL, Proctor SL and Dempsey A. Race and ethnicity in school psychology publications: A content analysis and comparison to publications in related disciplines. *Contemp Sch Psychol* 2013; 17(1); 129-142.
33. Sauser B and Boardman J. Taking hold of system of systems management. *Engng Mgmt J* 2008; 20(4): 44-49.
34. Cherns A. The principles of sociotechnical design. *Hum Rel* 1976; 29(8): 783-792.
35. Taylor J and Felten D. *Performance by design: Sociotechnical systems in North America*. Prentice-Hall, 1992.
36. Keating C, Sousa-Poza A and Mun J. System of systems engineering methodology. *National Centers for System of Systems Engineering Technical Paper*, 2004, pp.2-23.
37. Cloutier R, DiMario M and Gorod A. Net centricity and system of systems. In: Jamshidi M (ed.) *System of systems engineering – Innovations for the 21st century*. Hoboken, NJ: Wiley, 2009, pp.150-168.
38. Urbina A, Mahadevan S and Paez TL. Quantification of margins and uncertainties of complex systems in the presence of aleatoric and epistemic uncertainty. *Rel Engng Syst Saf* 2011; 96(9): 1114-1125.

Author biographies

Saikou Diallo is a research assistant professor at the Virginia Modeling, Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his MS in modeling and simulation (2006) and his PhD in M&S (2010) from ODU. His research focuses on the theory of interoperability as it relates to model-based data engineering and web services for M&S applications. He is currently the co-chair of the Coalition Battle Management Language drafting group, an M&S IEEE standard development group. His email and web addresses are sdiallo@odu.edu and https://www.odu.edu/directory/people/s/sdiallo, respectively.

Jose J. Padilla is a research assistant professor at the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. He received his PhD in engineering management from Old Dominion University in 2010. He is the head of the Human Dynamics Lab at VMASC. His email address is jpadilla@odu.edu, and his web page is https://www.odu.edu/directory/people/j/jpadilla.

Yiannis Papelis is a research professor at the Virginia Modeling, Analysis and Simulation Center of Old Dominion University. He received his PhD from the University of Iowa in 1993, a master's degree in electrical engineering from Purdue University in 1989, and a bachelor's degree in Electrical Engineering from Southern Illinois University in 1988. He is the virtual environments lead and the director of the Robotics Lab at VMASC. His email address is ypapelis@odu.edu, and his web page is https://www.odu.edu/directory/people/y/ypapelis.

Ross Gore is a research assistant professor at the Virginia Modeling, Analysis and Simulation Center of Old Dominion University. He holds a doctorate of philosophy (PhD) and a master's degree in computer science from the University of Virginia and a bachelor's degree in computer science from the University of Richmond. His research focuses on verification and validation as well as the mining of public data sources to inform models and simulations. His email address is rgore@odu.edu, and his web page is https://www.odu.edu/directory/people/r/rgore.

Christopher J. Lynch is a senior project scientist at the Virginia Modeling, Analysis and Simulation Center (VMASC) of Old Dominion University (ODU). He received a BS in electrical engineering from ODU in 2011, an MS in modeling and simulation (M&S) from ODU in 2012, and he is currently working on his PhD in M&S at ODU. He is currently a member of the Modeling and Simulation Interoperability (MSI) lab and the Human Dynamics Lab (HDL) at VMASC. His email address is cjlynch@odu.edu, and his web page is https://www.odu.edu/directory/people/c/clync008.