Human Interaction and Emerging Technologies (IHIET 2026), Vol. 236, 2026, 443–453
https://doi.org/10.54941/ahfe1008119

Systems-of-Systems: Engineering or Construct?

Pedro B. Agua, Anacleto Correia, and António Gonçalves

CINAV, Portuguese Naval Academy, Alfeite, Almada, Portugal

ABSTRACT

The concept of Systems-of-Systems (SoS) has become increasingly important across domains such as engineering, defence, healthcare, and digital infrastructure, describing complex assemblies of independent systems that collectively deliver capabilities beyond individual components. However, the concept remains debated: are SoS objective engineering entities or "constructs" used to interpret socio-technical complexity? Having its origins in the context of systems engineering and defence acquisition, SoS are defined by its operational and managerial independence, evolutionary development, and emergent behaviour. While these traits distinguish SoS from supersystems, they also reflect how complexity is framed. Two main perspectives are examined. The engineering perspective treats SoS as designable and controllable entities, emphasizing architecture, standards, governance, and lifecycle management approaches. Here, the challenge is extending traditional engineering methods to address scale, decentralization, and emergence. In contrast, the constructivist perspective sees SoS as interpretive frameworks for understanding loosely coupled, evolving systems shaped by organizational and social contexts. In this view, system boundaries, purposes, and identities are fluid and negotiated. Each perspective has implications. Engineering approaches support structure and measurable performance but may overlook human and political dynamics. Constructivist approaches highlight power, incentives, and adaptation but offer less practical design guidance. The paper argues that real-world SoS embody both views, combining engineered structures with emergent, socially shaped dynamics. It proposes a synthesized perspective that understands SoS as socio-technical phenomena—partly designed and partly constructed. This dual view suggests that effective SoS design and management require integrating engineering practices with insights from organizational theory, complexity science, and systems thinking.

Keywords: Agreement mechanisms, Construct, Systems-of-systems, Systems engineering, Socio-technical systems

INTRODUCTION

The transformation of engineered systems over recent decades has been both profound and largely under-theorised within engineering disciplines when it comes to the concept of systems-of-systems (SoS). Traditional systems engineering emerged in a context where systems could be bounded, specified, and controlled – aka designed. Large-scale projects such as aerospace systems, defence platforms, and industrial infrastructure could be decomposed into subsystems, engineered independently, and integrated through carefully managed interfaces. This obvious industrial-age mindset has at its core

Received May 10, 2026; Revised June 15, 2026; Accepted July 5, 2026; Available online August 25, 2026

a paradigm which assumed that complexity could be managed through decomposition and that system behaviour could be predicted through analysis – aligned with a "command and control" mindset.

However, the systems that now dominate engineering practice differ fundamentally from those that gave rise to systems engineering as a discipline. Contemporary systems are increasingly interconnected, distributed, and interdependent – and when it comes to SoS, independent and self governing. Capabilities are no longer delivered by single systems but by configurations of multiple systems interacting across organisational, geographical, and technological boundaries. These Systems-of-Systems are not merely larger systems; they represent a qualitatively different form of complexity. For instance most modern defence capabilities depend on such configurations, where independent systems must interoperate to achieve operational goals (Dahmann, 2014; Dahmann and Henshaw, 2016). Yet the discipline of systems engineering has largely responded by extending existing methods rather than questioning their underlying assumptions. Systems-of-Systems Engineering is often presented as a natural evolution of systems engineering, incorporating additional considerations such as interoperability and architecture. Model-Based Systems Engineering further promises to address complexity through formal modelling and integration.

This paper argues that conventional systems engineering and Model-Based Systems Engineering approaches may be insufficient when applied to real-world Systems-of-Systems characterised by autonomy, distributed governance, evolutionary development, and emergent behaviour. Rather than being merely complicated, such systems exhibit properties associated with complex systems (Snowden and Boone, 2007; Castellani and Gerrits, 2021), where behaviour arises from interactions, evolves over time, and is difficult to fully predict or control. Therefore, they lie outside the domain for which traditional engineering methods were designed, and are effective – simple/clear and complicated context as suggested by Snowden and Boone (2007). This derives from the SoS defining characteristics, which make them very different from a broader system with subsystems, but still within a merely complicated context (Maier, 1998). To develop this argument, the paper adopts a theoretical critique grounded in complexity theory and the *Cynefin* framework (Snowden and Boone, 2007). It suggests that SoS occupy the complex domain, where deterministic design approaches are inappropriate. The paper further suggests that SoS should be understood as socio-technical constructs rather than engineered artefacts, and that engineering practice must shift accordingly. This papers is composed of five sections. After this introductory section, a background section provides relevant knowledge, which is followed by a brief section referring the used methods. A results section provides robust logical evidence, which is followed by a discussion and conclusion.

BACKGROUND

This section provides some relevant framing information regarding the concept of SoS and the hopeless features of traditional systems of systems approach in trying to design SoS, which may be further aggravated by Model based Systems Engineering, which poses an additional risk by providing the illusion of being able to design real world SoS. Moreover the Cynefin framework is briefly introduced as a valid approach in understanding the limitations of systems engineering for the task, elaborating in the concept of ‘construct’ as it is understood within the scope of this paper.

Systems-of-Systems and Their Defining Characteristics

The concept of Systems-of-Systems has been widely discussed, yet its implications remain insufficiently understood. Definitions typically emphasise the integration of independent systems to achieve higher-level capabilities. However, the defining characteristics of SoS reveal a deeper challenge, as they are under no centralised control design. For instance, NATO literature identifies operational independence and managerial independence as central features (Dahmann, 2014). These characteristics imply that constituent systems are not subordinate components but autonomous entities with their own objectives, lifecycles, and governance structures. This autonomy fundamentally alters the nature of system integration. Integration is no longer a technical process under central control but a negotiated outcome among independent actors, which only decide to integrate with a wider network of systems if they wish so, or as long as it is worth being part of the wider network. As an illustrative analogy, one may consider situations in which autonomous entities withdraw from a broader arrangement when perceived benefits no longer justify participation. However, such analogies should be interpreted cautiously, as socio-political systems are not directly equivalent to engineered Systems-of-Systems.

Evolutionary development further distinguishes SoS from traditional systems. Unlike engineered systems which are designed and delivered as complete entities, SoS evolve continuously. Systems are added, modified, or removed over time, often independently of the SoS itself. This results in a dynamic configuration – an “arrangement” - which cannot be fully specified at any point.

Emergent behaviour represents the most profound challenge. For instance as described in some NATO materials, emergent behaviour arises from interactions among system elements and cannot be predicted from their individual properties. This means that the behaviour of a SoS cannot be fully designed or controlled (Jamshidi, 2009). It must instead be observed and managed as it emerges – a sort of ‘Navigation on sight’ – and a clear evidence that in complex contexts the best one can do is probably learn how to navigate such complexity and not trying to control it.

The Failure of Traditional Systems Engineering

Traditional systems engineering is built on a set of assumptions that are incompatible with the characteristics of SoS. These assumptions include the existence of well-defined system boundaries, stable requirements, and predictable behaviour. For instance, Systems Engineering lifecycle approach, as codified in ISO/IEC 15288, reflects these assumptions, defining a structured progression from requirements to design, implementation, and validation.

Other approaches from a defence acquisition perspective, which typically have to deal with complicated systems fall into the same paradigm as the ISO/IEC 15288 (DoD, 2006; INCOSE, 2023).

In the context of SoS, these assumptions break down, as systems boundaries are fluid, as systems participate in multiple configurations and interact with their environment in complex ways. A clarifying example for an hypothetical fully connected system made of merely ten elements, would provide a clearer picture: in a fully connected system of n = 10 elements (a complete graph), there are 45 unique connections (edges). The total number of outcomes, or possible ways these connections can be configured (on or off), is 35,184,372,088,832 (2^{45}), with the exponent 45 being the number of links according to $n.(n-1)/2 = 10.(10-1)/2 = 45$. In contexts where the number of possible combinations (and outcomes) would run in the range of trillions it is hardly possible of establishing any cause-and-effect relationships, in order to apply an engineering design approach.

If one takes Ashby's Law of Requisite Variety which states that to effectively control a system (aka design a system), the number of controlling variables (or the variety of responses) available to the controller must be at least equal to the number of potential states or disturbances generated by the system intended to be controlled would mean having trillions of controlling variables, which can be potentially hypothesised, but hardly achievable and verifiable (Ashby, 1956). Moreover in a *de facto* SoS requirements are not stable but evolve as the system evolves, and obviously behaviour is not predictable nor stable but emerges from interactions.

The industrial-age mindset which is behind the assumption of central control is particularly problematic, as traditional systems engineering assumes that a system can be designed and managed by a single design authority. In a SoS context, control is distributed, and authority is often limited or absent. Decisions must be negotiated, at best through "agreement mechanisms" among stakeholders with different objectives and motivations, leading to outcomes that cannot be fully controlled, nor even predicted; again due to an insurmountable number of potential combinations or outcomes inherent to complex contexts.

Model-Based Systems Engineering: An Insufficient Response

Model-Based Systems Engineering has been proposed as a solution to the limitations of traditional systems engineering design, offering improved integration and analysis through formal models. However, MBSE remains fundamentally deterministic. It assumes that system behaviour can be represented and analysed through models, and that these models can inform design decisions. However, this assumption is not valid in the context of SoS, as most likely such a system to be relevant will be existing in a complex context – not merely clear or complicated ones (Snowden and Boone, 2007). The interactions that define SoS are often too complex, dynamic, and context-dependent to be captured in formal models and attempting to model a SoS in its entirety risks oversimplification and false confidence – posing a risk on itself. The model becomes a representation of an "imagined" system rather than the actual system. Moreover, MBSE relies on the concept of a "system of interest" with defined boundaries, while in SoS, boundaries are emergent and context-dependent. Systems may participate in multiple SoS simultaneously, and the configuration of the SoS may change over time in a dynamic, adaptive complex way. The notion of a fixed system of interest becomes untenable in the complex context.

The *Cynefin* Framework and the Nature of SoS Complexity

The *Cynefin* framework provides a critical lens for understanding why traditional engineering approaches fail in the context of SoS. Developed by Snowden and Boone (2007), the framework distinguishes between domains of order and complexity, each requiring different approaches to decision-making (Fig. 1 and Table 1). The Cynefin framework categorises domains as simple, complicated, complex, chaotic, and disorder. SoS reside in the complex domain, requiring probe–sense–respond strategies rather than analyse–design–implement.

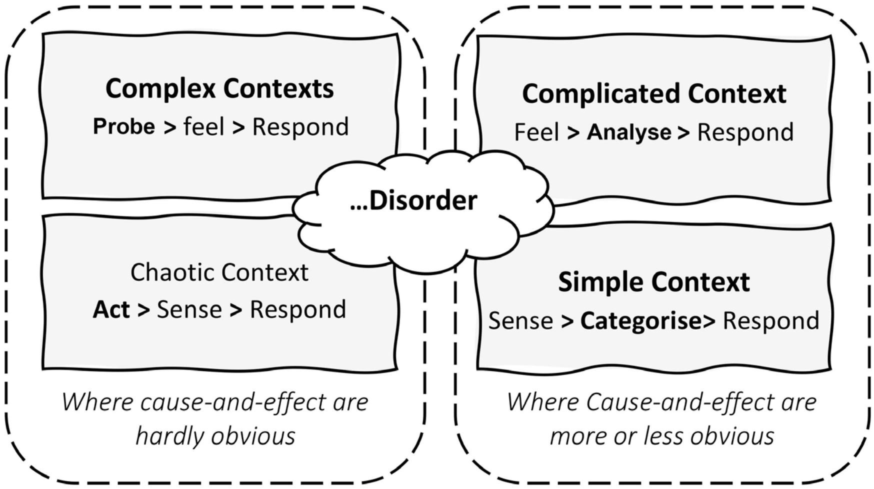

Figure 1: Cynefin framework [Adapted from Snowden and Boone (2007)].

Table 1: Distinguishing between a "construct" and traditional engineering design [Adapted from Snowden and Boone (2007)].

Categorization	Context	Obs.
Ordered World	Simple / Clear	Consistent events and easily recognizable patterns, in which cause-and-effect relationships are sufficiently clear and correct answers are common. Approach sequence: sense > categorize > respond
	Complicated	The domain of specialized diagnosis, in which cause-and-effect relationships exist but are not directly evident, and in which more than one correct answer is possible. Approach sequence: feel > analyse > respond

(Continued)

Table 1: Continued.

Categorization	Context	Obs.
Un-ordered World	Complex	Unpredictability is the norm, and there are no clear correct answers. Emergent behaviour is common. It needs creativity and innovative approaches to arrive at an appropriate response. Approach sequence: probe > feel > respond
	Chaotic	Characterized by great turbulence, in which the cause-and-effect relationships are not clear, nor are the answers entirely correct. It is the reality of a great time pressure that requires many simultaneous decisions. Approach sequence: Act > feel > respond

In the simple and complicated domains, cause-and-effect relationships are either obvious or discoverable through analysis. These domains correspond to the environments in which traditional engineering methods are effective. Systems can be designed, analysed, and controlled. The complex domain, however, is characterised by emergent behaviour and non-linear interactions. In this domain, cause-and-effect relationships are only apparent in retrospect. It is not possible to predict outcomes in advance. Instead, practitioners must probe the system, observe the results, and adapt accordingly. Systems-of-Systems clearly belong in the complex domain. Their behaviour emerges from interactions among independent systems and cannot be fully predicted. Attempts to apply deterministic design methods in this domain are not only ineffective but potentially harmful, as they create false expectations of control.

The Concept of "Construct" and a Potential Way Forward

A construct is a theoretical concept used to explain something abstract, while a traditional engineering design is a technical plan used to create something physical. The fundamental difference lies in observability and intent (Table 2).

Table 2: Distinguishing between a "construct" and traditional engineering design.

Feature	Construct	Traditional Engineering Design
Nature	*Abstract*. It exists as an idea or mental model (e.g.,"User Experience")	*Concrete*. It exists as a blueprint or CAD model for a physical object.
Primary Goal	To *define and understand* a complex phenomenon	To *specify and build* a functional solution.
Validation	Often proven through *statistical correlation* and indirect proxies (surveys, data).	Proven through *physics and math* (stress tests, thermal analysis).
Visibility	You cannot "touch" a construct; you can only measure its effects.	You can see the design and eventually touch the resulting product.

As a key distinction and as an analogy, a Construct (like "Structural Integrity") is the theory of how strong a building should be. It is an explanatory tool used to categorize different types of strength. Conversely "traditional engineering design" is the application—the specific calculations, material choices, and drawings that ensure that "Structural Integrity" is actually achieved in the real world. In short a construct tells what the idea is, while an engineering design tells how to make it work; yet in the context of systems-of-systems, due to the inherent complexity, traditional engineering design may not work, and risk providing a false sense of accomplishment, which poses a risk in itself. Moreover and crucial, evidence suggests that decentralized governance is not just an alternative but a necessity for Systems-of-Systems (SoS), as traditional centralized engineering models (Systems Engineering and MBSE) fail to account for the operational independence of constituent parts. In a SoS environment, individual systems are often owned and managed by different entities, making a "top-down" design impossible, and some authors have been pointing the associated challenges within such a paradigm of trying to do SoS engineering (Keating and Katina, 2011).

METHODS

This paper adopts a conceptual and theoretical methodology grounded in structured critical reasoning (Paul & Elder, 2009). It combines elements of soft systems thinking, abductive reasoning, and problem-structuring approaches to examine whether Systems-of-Systems should be treated primarily as engineered entities or as socio-technical constructs. These methods support analytical comparison and conceptual clarification rather than empirical validation. By combining a soft-systems methodology approach (Checkland, 1999), with the Theory of Constraints approach (Goldratt, 1990), and alternative analysis methods, such as logical thinking process (Dettmer, 2007), and problem solving methodologies (Chevallier, 2016) it provides a robust approach, as the use of multiple methods (aka models) cover the insufficiencies or blind spots of single models (Page, 2016). The reasoning approach was based on abductive thinking, as opposed to traditional inductive or deductive reasoning. Furthermore, the Five Whys technique is used to determine the root causes of the problem by asking "why" several times, while the Concept Mapping technique is used to produce tables depicting suggested relationships between concepts.

RESULTS

This paper has argued that Systems-of-Systems cannot be effectively addressed using traditional approaches. By making use of a fishbone diagram representation the problematic areas are exposed in graphical form (Fig. 2). This fishbone diagram comprises five major fields of issues: (1) Methods & processes, (2) Lack of Performance Metrics, (3) Environment & Evolution, (4) Management & Governance, and (5) Personnel & Culture. The arrows thickness are representative of the criticality of the parameters regarding the identified problem – *Failed or Unreliable SoS Engineering*.

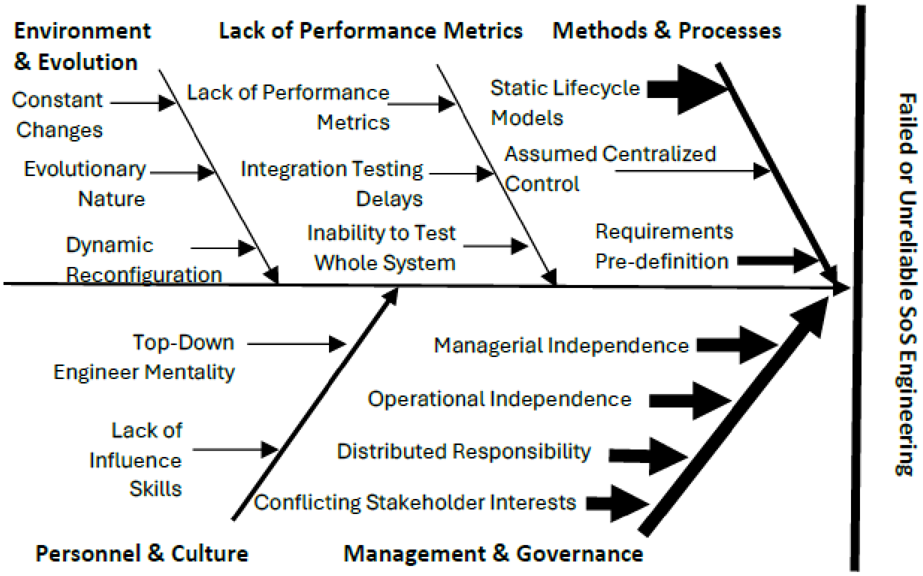

Figure 2: Ishikawa diagram helps with problem elicitation.

This fishbone diagram identified issues deserve some additional description, which taken the considerable number of issues and parameters is better accomplished by means of using a table representation (Table 3).

Table 3: Distinguishing between a "construct" and traditional engineering design.

Field	Issue	Obs.
Methods & Processes	*Static Lifecycle Models*	Rigid, phased approach (e.g. Waterfall) applied to systems that require continuous evolution
	Assumed Centralized Control	Mistakenly assuming one organization manages all constituent systems
	Requirements Pre-definition	Trying to define all SoS requirements upfront, ignoring that SoS are always in flux
Lack of Performance Metrics	*Lack of Performance Metrics*	No effective way to measure holistic SoS performance, only individual constituent systems' performance
	Integration Testing Delays	Independent systems developed at different speeds makes testing hardly predictable
	Inability to Test Whole System	Constraints prevent testing the complete SoS before deployment
Environment & Evolution	*Constant Changes*	Concepts become obsolete, and new systems enter the SoS during the process
	Evolutionary Nature	A finalized SoS is a myth; traditional systems engineering aims for a "finished" product
	Dynamic Reconfiguration	Inability to adapt to sudden changes in the environment, which forces the SoS to adapt

(Continued)

Table 3: Continued.

Field	Issue	Obs.
Management & Governance	*Managerial Independence*	Constituents have their own owners, goals, and funding, ignoring the SoS goals
	Operational Independence	Constituents operate independently and can leave the SoS at any time
	Distributed Responsibility	No single authority can dictate changes, technical or organizational
	Conflicting Stakeholder Interests	Different owners create conflicting goals for constituent systems
Personnel & Culture	*Top-Down Engineer Mentality*	Focusing on designing from scratch rather than negotiating integration
	Lack of Influence Skills	Engineers usually trained in direction, not negotiation

DISCUSSION

This paper has argued that Systems-of-Systems cannot be effectively addressed using traditional engineering approaches, which despite the hype of engineering fads, such as MBSE, are hardly able to be designed, for the reasons described in the preceding sections: (1) variability of combinations and potential emerging behaviours (outcomes); (2) the relevant SoS being normally in the Complex Context; and (3) the inherent independence, self-governance, and evolutionary characteristic of SoS.

Despite several authors promoting the idea that SoS can indeed be designed (Brook, 2014; Dahmann, 2014) given the definitions in previous sections – being a SoS autonomous, independent, continuously evolving, and keeping its own governance – there is hardly any chances of a central design authority (aka engineering) being able to design such entity. Some authors related with defence systems, a context where complicated systems are designed, often use arguments such as a frigate or an aircraft carrier are SoS. To clarify that such complicated systems are not a SoS as per the previous sections definition (including the features of autonomy, independent evolutionary path, and self-governing) it is helpful to clarify that there is no central authority, and as such no design is possible in the traditional terminology of systems engineering. - Does this mean that *one is helpless regarding the subject*? Obviously not, and we suggest that there are possible actions, but not the linear, industrial-age mindset control-based thinking typical of engineering approaches. The use of metaphors are sometimes helpful in explaining complex ideas, and an illustrative one would be the following: "the sea cannot be controlled, however one can learn how to navigate it"; *Mutatis, mutandis*, complexity is hardly controllable by design (if at all), but one can learn how to "navigate" such complexity. In the same sense that humankind has taken advantages of learning how to navigate the oceans, the same would be true regarding the advantages of learning how to navigate complexity, specifically by means of using applied complexity theories, and therefore approach SoS as arrangements or constructs. From the more traditional 'engineering approaches' Henshaw (2014, 2015) provide some useful insights, if one does not lose sight of SoS complex context and keeps it permanently into perspective.

The main argument behind this research and article is that a system-of-systems cannot be designed by using engineering thinking, but it is suggested that it is possible to set an 'arrangement' or 'construct' by means of agreement mechanisms which will keep the component systems tied together as along as the conditions which led to such agreement mechanisms are valid. Otherwise, not being this the case, one or more specific integrating systems may leave the broader SoS network if they do not see enough benefits of further belonging to the SoS – or, perhaps may decide a different "arrangement" from the one they had up to a certain moment in time – the UK relationship with European Union constitutes an example of such a situation.

CONCLUSION

This paper argues that many real-world Systems-of-Systems cannot be fully understood or managed through traditional deterministic engineering approaches alone. Their autonomy, distributed governance, evolutionary development, and emergent behaviour require complementary perspectives drawn from complexity theory, socio-technical analysis, and adaptive governance. Their defining characteristics place them in the complex domain, where linear deterministic approaches are inappropriate. The Cynefin framework provides a powerful lens for understanding this paradigm and highlights the need for new approaches based on experimentation, adaptation, and governance. Systems-of-Systems are not engineered, aka designed, artefacts but socio-technical constructs. Engaging with them requires a paradigm shift that represents one of the most significant challenges, and implies start considering applied complexity theories in a pragmatic way to construct, govern and manage within the context of SoS.

ACKNOWLEDGMENT

The authors would like to acknowledge the CINAV and the Portuguese Naval Academy for their support.

REFERENCES

Ashby, W.R. (1956). An Introduction to Cybernetics. London: Chapman & Hall.

Brook, P. (2014). "Architecting Systems of Systems", in Systems of Systems Engineering for NATO Defence Applications, NATO STO Educational Notes, STO-EN-SCI-276, Paper 2, NATO STO.

Castellani, B. and Gerrits, L. (2021). Map of the complexity sciences. Available at: https://www.art-sciencefactory.com/MapLegend.html (Accessed: 15 April 2026).

Checkland, P.B. (1999). Systems Thinking, Systems Practice. Chichester, UK: John Wiley & Sons Ltd.

Chevallier, A. (2016). Strategic Thinking in Complex Problem Solving. New York: Oxford University Press.

Dahmann, J.S. (2014).”Systems of Systems Characterization and Types”. NATO Publication Ref NBR EN-SCI-276-01.2015

Dahmann, J.S. and Henshaw, M.J. (2016). “Introduction to Systems of Systems Engineering”, INCOSE INSIGHT, October 2016, Volume 19, Issue 3, pages 12–16

Dettmer, H.W. (2007). The Logical Thinking Process: A Systems Approach to Complex Problem Solving. Milwaukee, WI: ASQ Quality Press

DoD (2006). System of Systems Systems Engineering Guide: Considerations for Systems Engineering in a System of Systems Environment. Washington, DC:Office of the Deputy Under Secretary of Defense for Acquisition

Goldratt, E.M. (1990). What is this Thing Called Theory of Constraints and how Should it be Implemented?. Great Barrington, MA: North River Press

Henshaw, M. (2015). ‘A Socio-Technical Perspective on SoSE’, in Systems of Systems Engineering for NATO Defence Applications 2015, AC/323 (SCI-276)TP/617. NATO STO

Henshaw, M., et al. (2014). ‘Systems of Systems Governance’, in Systems of Systems Engineering for NATO Defence Applications. NATO STO Educational Notes, STO-EN-SCI-276

INCOSE (2023). Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. 5th edn. Edited by D.D. Walden. Hoboken, NJ: Wiley

Jamshidi, M. (2009). Systems of Systems Engineering – Innovations for the 21st Century. Hoboken, NJ, USA: Wiley and Sons

Keating, C.B. and Katina, P.F. (2011). ”Systems of systems engineering: Prospects and challenges for the emerging field”, International Journal of System of Systems Engineering, 2(2/3), pp. 234–256. doi: 10.1504/IJSSE.2011.040556

Maier, M.W. (1998). “Architecting Principles for Systems-of-Systems”. Systems Engineering, 1(4).

Page, S.E., (2018). Why “Many-Model Thinkers Make Better Decisions”. Harvard Business Review. 19 November. Available from: https://hbr.org/2018/11/why-many-model-thinkers-make-better-decisions [Accessed 15 April 2026].

Paul, R. and Elder, L. (2009). The Miniature Guide to Critical Thinking: Concepts and Tools, 6th Ed. The foundation for Critical Thinking, Foundation for Critical Thinking, Santa Barbara, California

Snowden, D.J. and Boone, M.E. (2007). “A leader’s framework for decision making”. Harvard Business Review, 85(11), pp. 68–76, 149