



## A view on Systems of Systems (SoS)

Mohamad Ali Assaad, Reine Talj, Ali Charara

### ► To cite this version:

Mohamad Ali Assaad, Reine Talj, Ali Charara. A view on Systems of Systems (SoS). 20th World Congress of the International Federation of Automatic Control (IFAC WC 2017) - special session, Jul 2016, Toulouse, France. hal-01741416

**HAL Id: hal-01741416**

**<https://hal.science/hal-01741416>**

Submitted on 23 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A view on Systems of Systems (SoS)

Mohamad Ali Assaad, Reine Talj, Ali Charara \*

*\* Sorbonne universités, Université de technologie de Compiègne,  
CNRS, laboratoire Heudiasyc, UMR 7253, 57 avenue de Landshut,  
60203 Compiègne Cedex France.  
(e-mail: mohamad-ali.assaad, reine.talj, ali.charara@hds.utc.fr).*

---

**Abstract:** This paper presents literature review on Systems of Systems (SoS), a type of complex systems that are being more recognised for their importance recently. SoS exist across multiple domains, and thus multiple visions of what they are and how they are used. Traditional systems engineering is not enough to address the challenges of SoS because of their complexity. The solution however to build and manage SoS is through an architecture framework that contains the “invariants” of the SoS in question, and the set of rules that manage the interactions between components.

*Keywords:* Systems of systems, complex systems, emergence, interaction, architecture framework.

---

## 1. INTRODUCTION

Throughout the evolution of systems, the engineering effort dedicated to hardware vs. software have had a dramatic shift. At the start the effort was mainly on hardware development, with a little focus on software, whereas now the software part is dominant [Maier et al. (2000)]. Along side this shift, the value of information is in a constant increase, and communication between different systems has led to an emergence of a new types of systems, that rose from the collaboration between independent systems, that share a common objective. This new type of systems is usually referred to as *Systems of Systems (SoS)*, and it exists in almost every domain of applications [Mohsen et al. (2016)].

Systems of Systems offer capabilities that are not achievable practically in a single system, they have no single point failure, and continue to work even in a dynamical environment. So far there is no universally accepted definition of SoS, however these systems share characteristics that distinguish them from other complex systems.

This paper presents some of the work done in the literature on SoS, Section 2 lists some of the different definitions, with a special definition based on the characteristics of SoS and its components, Section 3 contains some of the most important characteristics of SoS, divided into two categories: SoS characteristics on the upper level, and Components characteristics, on the lower level. Even though SoS share the same characteristics, some attributes change from system to another, that’s why, based on objectives, governance, and the inter-relationship between components, SoS are classified into four types, listed in Section 4. An important question arises whether it is possible, facing the new challenges of SoS, to use traditional methods to build and manage SoS, Sections 5 and 6 help

answering to this question, by showing the differences between traditional systems engineering requirements, and SoS engineering requirements, and introduce the principle of architecting of a SoS. Section 7 introduce the concept of frameworks, and lists some of the used frameworks in SoS. Finally, Section 8 closes this paper with a conclusion.

## 2. SOS DEFINITION

Systems of systems can be found in almost every domain of application, in health-care [Hate (2009)], defense [DoD (2008)], transportation [Maier. (1997)], etc..

This diversity proved to be a problem, since every domain has its own definition for SoS. Some of these definitions are:

**Definition 1:** Enterprise systems of systems engineering is focused on coupling traditional systems engineering activities with enterprise activities of strategic planning and invesment analysis [Carlock and Fenton (2001)].

**Definition 2:** SoS integration is a method to pursue development, integration, interoperability, and optimization of systems to enhance performance in future battlefield scenarios [Pei (2000)].

**Definition 3:** A SoS is every object that biology studies. [Jacob (1974)].

The above definitions mainly explain the use of SoS in specific domains, and cannot be used to establish a general definition of SoS.

Another type of definitions exists, which describe SoS regardless of the domain. Below we state a few:

**Definition 4:** A set of several independently acquired systems, each under a nominal systems engineering process; these systems are interdependent and form in their combined operation a multifunctional solution to an overall coherent mission [Eisner et al. (1991)].

---

\* This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program “Investments for the future” managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

**Definition 5:** A large widespread collection or network of systems functioning together to achieve a common purpose [Shenhar (1994)].

**Definition 6:** SoS are large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal [Jamshidi (2008)].

**Definition 7:** Assemblage of systems that can be potentially acquired and/or used independently, for which the designer, the acquirer and/or the user tries to maximize the performance of the global value chain, at a given time and for a set of foreseeable assemblages [Luzeaux (2013)].

Assume we have a system  $X$ , the interest in definitions is to answer this question: *Is  $X$  a System of systems, or not?* The latter definitions are more useful in answering this question. However, one definition stands out from the others, which is based on the characteristics of the components of the SoS, comes from [Maier (1996)], in which he states:

**Definition 8:** A system-of-systems is an assemblage of components which individually may be regarded as systems, and which possesses two additional properties:

**Operational Independence of the Components:**

The component systems must be able to usefully operate independently.

**Managerial Independence of the Components:**

The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems.

### 3. SOS CHARACTERISTICS

This diversity of SoS, pose a problem when trying to find a common definition.

However, as a starting point, a quick analysis of the different definitions shows us that a SoS can be divided into two parts: the global system, which is the result of the gathering of the components, and the components themselves.

This division is important for us, because it will help us tell if  $X$  is a SoS or not. How? Well even though the study of SoS is still immature, but we are surrounded by them, and by studying such systems, we actually can infer some characteristics of both parts (SoS and components).

Below a list of some of the characteristics of SoS and its components, they were gathered from [Maier (1996)], [DiMario (2010)] and [Gorod et al. (2008)].

#### 3.1 Components Characteristics

**Autonomy:** The autonomy of components encloses both operational and managerial independence. In another word, the components can and will choose their decisions based on their interests.

**Heterogeneity:** A direct consequence of the “managerial independence” mentioned in Definition 2. The components must be managed by different parties.

**Belonging:** The components must belong to the SoS based on their choice, even though this belonging will lose them part of their autonomy, but this loss will be compensated by the wins they will get from the SoS, which is the exact reason why they choose to belong.

SoS Objectives: Emergent					SoS Objectives: Recognized
SoS Governance: Community	Virtual	Collaborative	Acknowledged	Directed	SoS Governance: Centrally Managed
Inter-relationship: Independent					Inter-relationship: Subordinate

Fig. 1. The four types of SoS [Colling et al. (2016)]

#### 3.2 Global Characteristics

**Evolution:** The autonomy of the components, means that they have the ability to evolve and change, regardless of the SoS. On the other hand, SoS work in an unpredicted, dynamic environment. All of that, in addition to the fact that a SoS’s objectives change as well, leads to an ever evolving system, that must adapt to account for internal, together with external changes.

**Emergence:** One of the most important characteristics of a SoS, and is inherited from complex systems. Simply: emergence is a behaviour of the global system, which is the result of the interaction between components, and is not a property of any of the components.

**Diversity:** SoS should offer a lot of functionalities,

## 4. SOS TYPES

The above characteristics may be used to classify a system as SoS or not. However, Maier in [Maier (1996)], goes beyond that and proposed three types for SoS, based on their objectives, governance, and the inter-relationship between the components. The three different types are:

**Directed:** The system is built for a specific purpose, and centrally managed. An example would be the systems responsible of the development of the Future Combat Systems in the US Department of Defense (DoD) [Dahmann et al. (2008)].

**Collaborative:** In this case, the components collaborate to fulfill the central objectives. In collaborative systems, the central management does not have the coercive power to run the system. An example would be the Intelligent Transport Systems [Maier. (1997)].

**Virtual:** Virtual SoS emerges from the interaction between components, whereas the objectives are unknown, and there is no central authority. The system is maintained through invisible mechanisms. National economies can be classified as Virtual SoS. [Maier (1996)].

However, later, Dahmann et al. (2008) proposed a fourth type, *Acknowledged SoS*, which shares attributes from both Directed and Collaborative SoS. *Acknowledged SoS* were found in a lot of the DoD’s applications. These systems have a central management and resources for the SoS, but the components retain their full independence, and they are responsible of the changes in the SoS.

Fig. 1 shows the different types of SoS based on the following three attributes: Objectives, Governance, and the inter-relationship between components.

	SE	SoSE
<b>Focus</b>	Single Complex System	Multiple Integrated Complex Systems
<b>Objective</b>	Optimization	Satisfaction, Sustainment
<b>Boundaries</b>	Static	Dynamic
<b>Problem</b>	Defined	Emergent
<b>Structure</b>	Hierarchical	Network
<b>Goals</b>	Unitary	Pluralistic
<b>Approach</b>	Process	Methodology
<b>Timeframe</b>	System Life Cycle	Continuous
<b>Centricity</b>	Platform	Network
<b>Tools</b>	Many	Few
<b>Management Framework</b>	Established	?

Fig. 2. Systems Engineering vs. SoS Engineering [Gorod et al. (2008)]

## 5. SYSTEMS ENGINEERING VS. SOS ENGINEERING

So far, we have seen that SoS are systems that have some characteristics that differentiate them from other types of systems. However, do these differences pose a problem if we try to build a SoS using traditional systems engineering (SE) principles?

Comparisons between traditional systems and SoS of [Rebovich. (2008)], [Gorod et al. (2008)], and the comparison between systems and *Acknowledged SoS*, shows that the current system engineering (SE) process is not enough to build and manage SoS.

Fig. 2 from [Gorod et al. (2008)] shows the major differences between SE and SoSE.

## 6. ENGINEERING VS. ARCHITECTING

The difficulties facing SoS come mainly from the fact that the components are independent, this means that they may undergo changes during their lifetime and during the run-time of the SoS they belong to, which were not foreseen when building the SoS. In addition to this, the objectives of SoS are always changing, undesirable emergent behaviors could appear in the system, the environment in which SoS are working is in constant change as well.

All of these difficulties, and much more, must be considered in design-time of the SoS. We still lack the engineering tools to address these problems, on the other hand, architecture offers more nonquantative tools that can help.

In Maier et al. (2000), the author explains the differences between *Engineering* and *Architecting* a system: “*Engineering* is a deductive process that deals with measurables using analytical tools, whereas *Architecting* on the other hand is an inductive process that deals with unmeasurables using nonquantative tools and “guidelines” based on practical lessons learned”.

These “guidelines” can be inferred from case studies performed on existing SoS (e.g. The internet, telephone networks) [Maier (2005)], and implemented in a framework that guides the construction of new SoS. The following section provides a brief description of frameworks, and lists some of the existing frameworks that were created for SoS, or being used to build SoS.

## 7. SOS FRAMEWORK

Complex evolving systems evolve more effectively and more rapidly if they have stable intermediate forms (invariants) [Rechtin (1991)]. This means that the SoS should be capable of operating before full deployment or construction is achieved [Maier (1996)], also means that the SoS should still operate even if some components left that system.

These “Invariants” play an important role in the sustainability and evolving of SoS. Another important aspect of SoS is the communication between components, since components are managerially independent, the collaboration usually is done via information sharing between components. In [Maier (1998)], the author extracted four heuristics that serve as principles for architecting SoS:

- Stable intermediate forms: The system must have a continuous operation, even if it faces a sudden change in its configuration.
- Policy triage: Guidance in selection, support, and rejection of the components, in addition to standards.
- Ensuring collaboration: The reasons that induce components to collaborate must be designed into the system.

The fourth heuristic, “Leverage at the interfaces” is the most important when architecting a SoS, in a sense that the only way the authority can control the SoS behavior is by controlling how the components collaborate. This is done using a framework that contains rules on how, when, what, and with whom a component must communicate.

An architecture framework is: “a tool [...] that should describe a method for designing an information system in terms of a set of building blocks, and for showing how the building blocks fit together [...]. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks.” [TOGAF (2006)]

So, an architecture framework should contain a detailed description of the objectives of SoS, the components, the inter-relationship between them, and the rules that govern the SoS in question (the invariants). In addition, due to the complexity of SoS and the detailed description needed from a framework, it is hard to describe all the aspects of a SoS in a single-view framework [Cole (2009)], that’s why a good framework for SoS should use a multiview approach. Multiple frameworks exist, that are being used to manage and architect SoS, some of them are specifically made for SoS, others originally created for other purposes, are being used in SoS.

Despite their differences, SoS face similar challenges that need to be addressed by any framework that targets SoS. To benefit from this similarity, case studies on already established SoS, like the internet, the telephone networks,

etc., may help the designer to find solutions to some problems in his SoS, or may shed the light on unforeseen problems that might emerge in the future. [Gorod et al. (2014)] contains multiple case studies on systems that were built as SoS, and some lessons learned.

Some of the frameworks used for SoS are:

- The Zachman Framework [Zachman (1987)]
- The Department of Defense Architecture Framework (DoDAF) [DoD (2007)]
- The Ministry of Defense Architecture Framework (MoDAF)
- System of Systems Operational Management Matrix (SoSOMM) [Gorod et al. (2007)]

The Zachman framework, originally made for enterprise architecture, was the first architecting framework with multiview descriptions, and was the inspiration of all the other frameworks [Cole (2009)]. The DoDAF is the framework used across the defense department in the United States, while the MoDAF is the one being used in the United Kingdoms, they have both a multiview approach, however the viewpoints are slightly different. The SoSOMM is a framework proposed in [Gorod et al. (2007)], which uses some of the best practices of network management and applies them along side some SoS guides, in order to build and manage SoS. SoSOMM still however needs practical validation [Gorod et al. (2007)].

## 8. CONCLUSIONS

System of systems is a new type of systems that is formed from the collaboration between its components, which by themselves are independent systems. SoS are applicable in almost any domain, and are known by their emergent behavior. Based on their objectives, governance, and the inter-relationship between components, SoS can be classified into four types: Directed, Acknowledged, Collaborative, and Virtual; In real applications however, a SoS may share attributes from multiple types.

Classical systems engineering is not enough to address the challenges facing SoS. What is important in a SoS are invariants and communication between components. A good way to manage them in a SoS is a good “architecture framework”. Multiple frameworks exist, some are specifically made for SoS, others are from network management, or enterprise management.

Our understanding for SoS is growing, however a lot have to be done. A lot of challenges can’t be solved with the current methods and need specific SoS methods.

Systems Engineering Body of Knowledge contains a special part about SoS, in which the interested reader can find a lot of important references on SoS

([http://sebokwiki.org/wiki/Systems\\_of\\_Systems\\_\(SoS\)](http://sebokwiki.org/wiki/Systems_of_Systems_(SoS))).

## REFERENCES

Mosleh Mohsen, Peter Ludlow, and Babak Heydari. Distributed Resource Management in Systems of Systems: An Architecture Perspective. *Systems Engineering* 19.4 (2016): 362-374.

Yutaka Hata, Syoji Kobashi, and Hiroshi Nakajima. Human Health Care System of Systems. *IEEE SYSTEMS JOURNAL*, VOL. 3, NO. 2, JUNE 2009.

The American Department of Defense (DoD). *Systems Engineering Guide for Systems of Systems*. 2008.

Mark W. Maier. Research Challenges for Systems-of-Systems. 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005, vol. 4, pp. 3149-3154.

Carlock, P. G., and R. E. Fenton. Systems of Systems (SoS) enterprise systems for information-intensive organizations. *Systems Engineering* 4(4):242-261, 2001.

Pei, R. S. Systems of Systems Integration (SoSI) - a smart way of acquiring Army C4I2WS systems. *Proceedings of the Summer Computer Simulation Conference*, Monterey Bay, CA, June 2-4, 2000.

F. Jacob. *The Logic of Living Systems*. London, U.K.: Allen Lane, 1974.

H. Eisner, J. Marciniak, and R. McMillan. Computer-aided system of systems (C2) engineering. Presented at the IEEE Int.Conf. Syst., Man Cybern., Charlottesville, VA, 1991.

A. Shenhar. A new systems engineering taxonomy. *Proc. 4th Int. Symp. Nat. Council Syst. Eng.*, 1994, pp. 261-276.

M. Jamshidi. *Systems of Systems Engineering: Principles and Applications*. CRC Press, Ch. 1, Nov. 2008.

D. Luzeaux. *Systems of Systems*. John Wiley & Sons, Inc. 2013.

M. Maier. *Architecting principles of systems-of-systems*. Presented at the 6th Ann. Int. Symp. Int. Council Syst. Eng., Boston, MA, 1996.

M. J. DiMario. *System of Systems Collaborative Formation*. World Scientific Publishing Co. Pte. Ltd, 2010.

Alex Gorod, Brian Sauser, and John Boardman. *System-of-Systems Engineering Management: A Review of Modern History and a Path Forward*. *IEEE SYSTEMS JOURNAL*, VOL. 2, NO. 4, DECEMBER 2008.

Dahmann, Judith and Kristen Baldwin. *Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering*. Montreal, Canada: IEEE Systems Conference, 7-10 April, 2008.

M.Maier. On architecting and intelligent transport systems. *Joint Issue IEEE Trans. Aerosp. Electron. Syst./Syst. Eng.*, vol. 33, pp. 610-625, Apr. 1997.

Bernie Collins, Dr. Steve Doskey, Dr. James Moreland. *Relative Comparison of the Rate of Convergence of Collaborative Systems of Systems: A Quantified Case Study*. Presented 10 May 2016 to the System of Systems Engineering Collaborators Information Exchange, George Washington University.

George Rebovich, Jr. *Systems of Systems Engineering: Principles and Applications*. CRC Press, Ch. 6, Nov. 2008

Mark W. Maier, Eberhardt Rechtin. *THE ART OF SYSTEMS ARCHITECTING*. 2nd edition, CRC Press, Boca Raton, FL, 2000.

Rechtin, E. *Systems Architecting: Building Complex Systems*. Prentice Hall, 1991.

M. Maier. *Architecting principles for systems-of-systems*. *Systems Engineering*, 1:4, pp 267-284, 1998.

The Open Group Architecture Framework. <http://pubs.opengroup.org/architecture/togaf8-doc/arch/>.

- Reggie Cole. SoS Architecture. In “Systems of Systems Engineering: principles and applications”, edited by Mo Jamshidi, Boca Raton, FL, USA: CRC Press, 2009.
- Zachman, J. A framework for information systems architecture. IBM Systems Journal, 1987.
- United States Department of Defense. DoD Architecture Framework Version 2.02. United States Department of Defense, 2010.
- Gorod, A., Gove, R., Sauser, B., Boardman, J. System of Systems Management: A Network Management Approach. Presented at the IEEE Int. Conf. Syst. Syst. Eng., San Antonio, TX, 2007.
- A. Gorod, B.E. White, V. Ireland, J.S. Gandhi, B. Sauser. Case studies in System of Systems Enterprise systems and Complex Systems Engineering. in I have simply, 2014, CRC Press.