

## Characterizing Systems of Systems change and failure via network-based metrics

Sarra Fakhfakh<sup>1,2</sup>, Andreas-Makoto Hein<sup>1</sup>, Marija Jankovic<sup>1</sup>, Yann Chazal<sup>2</sup>

<sup>1</sup> Université Paris-Saclay, CentraleSupélec, Laboratoire Génie Industriel, France

<sup>2</sup> Renault SA, France

**Abstract:** A System of System (SoS) is a synthesis of independent systems functioning together towards a common goal. They are characterized by their dynamic nature and evolvability during operation: addition, removal, and modification of component systems and functions. It is, therefore, important to characterize the tolerance of such systems to changes and failures. Most change propagation and failure analysis methods require some knowledge of failure and change probabilities, failure modes, and design parameters, which is difficult to obtain or unavailable to an SoS decision-maker, as component systems are independent in their management and operation. Consequently, this paper uses high-level SoS functional models and network-based metrics to characterize SoSs functions and assess the functional change and failure of such systems. The proposed measures are deployed on an electric vehicle to grid-related service to show how it can aid an SoS decision-maker during the system's development and operation.

*Keywords:* Systems of Systems, Functional modelling, DSM, Network-based metrics

### 1 Introduction

New mobility services such as Electric Vehicle to Grid (EV2G), connected mobility, and autonomous vehicle services are becoming more and more important and complex. Such services are supported by systems (e.g., electrical grids and electric vehicles) and are developed, managed, and operated by various stakeholders (e.g., energy providers, car manufacturers, and service providers). These systems can be considered as Systems of Systems (SoSs). SoSs are defined as “a class of systems which are built from components which are large scale systems in their own right” (Maier, 1996). SoS are characterized by the operational and managerial independence of their elements, their evolutionary development, emergent behavior, geographic distribution (Maier, 1996), as well as the interoperability, complementarity, and holism of their Component Systems (CSs) (Keating & Katina, 2011). In essence, SoSs are a synthesis of distributed, heterogeneous, and independent CSs collaborating and functioning together towards a common goal (Petitdemange, Borne, & Buisson, 2018; Uday & Marais, 2015). By nature, SoSs are evolvable, and their design is dynamic as CSs and functions are added, removed, or modified during operations and runtime (Mohsin, Janjua, Islam, Vicente, & Neto, 2019; Petitdemange et al., 2018). Furthermore, SoSs operate in highly uncertain environments (e.g., new requirements or changing stakeholder needs) (Uday & Marais, 2015), leading to changes and failures in CSs functioning. Therefore, it is important to characterize the tolerance of such systems to changes and failures.

Various change and failure analysis methods have been proposed in the systems engineering literature. Failure mode and effect analysis (FMEA), Fault Tree Analysis (FTA), or Risk in Early Design (RED) are widely used failure analysis methods (Walsh, Dong, & Tumer, 2018). Among others, the Change Prediction Method (CPM) (Clarkson, Simons, & Eckert, 2004) is a significant reference for change propagation analysis (Sarica & Luo, 2019). However, these methods require some knowledge of failure likelihood, change probabilities, and design parameters. Knowledge of changes and failures (e.g., probabilities) is difficult to obtain or unavailable for an SoS decision-maker due to the dynamic and evolving nature of SoS. Moreover, the CSs' design parameters and detailed descriptions are usually unknown to an SoS decision-maker because he/ she usually develops, manages, or operates a specific CSs. Besides, the mentioned methods focus on independent failures or changes and do not necessarily consider life cycle dependencies. In addition, they are computationally costly (Sarica & Luo, 2019; Uday & Marais, 2015; Walsh et al., 2018). These limitations are significant in the context of SoS development (Sarica & Luo, 2019) as (Luo, 2015) showed that design dependency cycles between components are one of the main causes of product evolvability.

The use of network-based metrics to analyze system change and failure propagations (Haley, Dong, & Tumer, 2016; Sarica & Luo, 2019) is increasingly recognized to address the gaps of usual methods. In this paper, we propose to use network-metrics to characterize functional SoSs change and failure. For this purpose, we build a network of SoS functions based on high-level SoS functional descriptions.

Section 2 reviews the use of network-based metrics in characterizing systems change and failure propagation. In section 3, we show how to build a functional network based on SoS functional chains and how network-based metrics can be used. We deploy these measurements on an electric vehicle-related service use case in section 4. Finally, we conclude in section 5 and provide future research perspectives.

## 2 Literature Review

The previously presented methods for analyzing changes and failures are rather limited when addressing SoS development. For SoS development, we argue that a method would need to have the following characteristics:

- *(C1) No knowledge of failure likelihood, change probabilities required:* Knowledge on failure likelihood and change probabilities is difficult to obtain in the context of SoSs development as CSs, and their functions change and evolve during runtime.
- *(C2) No knowledge of design parameters required:* Since CSs are independent, design parameters are not necessarily available to an SoS decision-maker.
- *(C3) Consider cycle dependencies:* Luo, (2015) showed that intercomponent design dependency cycles in system architecture give rise to product evolvability. As such, dependency cycles concern SoSs (Sarica & Luo, 2019).
- *(C4) Low computational cost*

In the following, we review network-based change and failure analysis methods. Table 1 shows how the network metrics they use address partly or fully these limits (Sarica & Luo, 2019; Uday & Marais, 2015; Walsh, Dong, & Tumer, 2019), and as such, can be relevant in the context of SoSs development.

Representing complex engineering systems as a network of interconnected components, node centrality metrics such as degree centrality and eigenvector centrality are used to characterize the tendency of a node (e.g., component) to propagate changes and failures. For example, Sosa et al., (2011) propose a method to identify hubs (defined as highly connected components) using product Design Structure Matrices (DSMs) and measuring node degrees. The authors empirically show that the presence of hubs in system architectures is associated with a low number of defects in response to changes. Chai et al., (2011) considered a networked infrastructure system in the context of oil and gas industries and used degree centrality measures to identify infrastructures that are most relied upon and, as such, might cause most significant cascading failures. To consider cyclic dependency between components, Sarica & Luo, (2019) use eigenvector centrality measures to characterize the influence and susceptibility of components to change. Eigenvector calculations-based measures have also been used by (Li, Wang, Zhong, & Zou, 2018) to identify influential function modules by considering function modules as the nodes of the network.

Other studies explore network metrics to characterize the overall robustness of a system (or an SoS) to the removal or loss of a node or an edge (e.g., a CS removal or an interaction loss between two CSs). For instance, Antul et al., (2017) represents an SoS as the network of its CSs and measures the algebraic connectivity to capture “a network’s vulnerability to disconnection (e.g., removal of a CSs).” In a similar fashion, the robustness coefficient has been used in the literature (Haley et al., 2016; Papanastasiou, Duffy, Whitfield, Knight, & Robb, 2020; Walsh et al., 2019) and characterizes the largest connected component (connected nodes) after a node removal (e.g., removal of a CS in an SoS or the loss of a function). Papanastasiou et al., (2020) use the robustness coefficient to compare naval distributed engineering system architectures options. Walsh et al., (2019) use the robustness coefficient to explore the correlation between robustness and modularity. The Average Shortest Path Length (ASPL) is another metric that has been used to characterize the robustness of a network (a system) (Walsh et al., 2019). The ASPL measures the average shortest distance between two nodes in the network. As such, the ASPL describes the relative efficiency of a flow to travel throughout a network. The ASPL has been usually used to characterize the robustness of the overall network and compare nominal architectures and failed cases (after node removal) (Haley et al., 2016). In (Walsh et al., 2018), a variation in the ASPL is used to characterize the relative vulnerability of each node locally. A “vulnerable” node is defined as a node whose removal disconnects a large portion of the network or increases the ASPL. To evaluate the vulnerability of a system parameter, the authors measure the variation of the ASPL ( $\Delta$ ASPL) of a behavioral network (i.e., design parameters network) after the parameter node failure. The authors express a failure by decreasing the weights of all edges associated with that node. As such, the higher the  $\Delta$ ASPL is, the more vulnerable the parameter node is.

Succinctly, network-based change and failure analysis methods rely on local network-metrics (network centrality metrics) characterizing the tendency of each node to propagate change or failure, or global network-metrics characterizing the robustness of the overall network to changes or failures (e.g., Robustness Coefficient) (Table 1). Global network-metrics (ASPL) have also been adapted to characterize the local vulnerability of nodes (Walsh et al., 2018). Local and global metrics are complementary and address different purposes. Hence, both global and local network-metrics can be useful tools for analyzing

and comparing different SoS architectures giving insights to an SoS decision-maker during the system's development and evolution.

Table 3: Characteristics of the most used network metrics in change and failure analysis methods

<i>Network Metrics used in change and failure propagation methods</i>		<i>Measures characteristics</i>			
		<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>
<i>Local metrics</i>	<i>Degree centrality</i>	x	x (depending on the considered network)		x
	<i>Eigenvector centrality</i>	x	x (depending on the considered network)	x	x
<i>Global metrics</i>	<i>Algebraic connectivity</i>	x	x (depending on the considered network)	x	x
	<i>Robustness coefficient</i>	x	x (depending on the considered network)		x
	<i>ASPL</i>	x	x (depending on the considered network)		x

The various studies of the literature use networks with different views of the systems, leading to different interpretations of the network metrics. The authors mostly consider networks of interconnected components (or CSs for SoSs). The interconnection or dependency between components might model spatial, structural, material, energy, or information dependencies. A component dependency is represented by a weighted or unweighted edge, depending on the intensity or type of dependency (Sarica & Luo, 2019). More recent publications considered networks of function modules built based on functional chains (Li et al., 2018) or behavioral networks based on mathematical details of the system's governing equations Walsh et al., (2018) (in which case the design parameters are required C2 Table 1). Thus, considering such networks in the context of SoS is worth exploring since they have rarely been considered in the literature.

The use of network-metrics is a promising avenue to analyze changes and failures in SoS development. However, SoS changes and failure include CSs and functionalities addition, removal, or modification during runtime, which distinguishes them from monolithic systems. Therefore, the interpretability of measures and results highly depends on both the systems specificities and the network used to model them. As most SoS network-based change and failure analysis methods consider a network of the CSs, we propose an approach to build an SoS functional network and justify its use. We also propose a use and an interpretation of local network-metrics to analyze such a network.

### 3 An approach for Systems of Systems functional change and failure characterization

#### 3.1 SoS functional dependency matrix and network definition

In the following, we focus on functions and their dependencies as the main object of analysis, due to their practical relevance and the availability of SoS frameworks that are based on this perspective. From a practical perspective, an SoS decision-maker usually develops, manages, and operate a specific CS, or manages the overall SoS from a high-level perspective. Hence, SoS decision-makers do not have extensive knowledge on the structure of all CSs or their design parameters. They rather have a high-level representation of how CSs function together to achieve a common goal, mission, or service during development and operation. From a theoretical perspective, Luo, (2017)puts functions and

functionality at the heart of SoS innovation and design. SoS development is described as the expansion of the thinking beyond the boundaries of independent systems and how the combination of the functions of individual systems can be synthesized to create a new functionality. Different SoS frameworks include functional representations for SoS (Fakhfakh, Hein, Chazal, & Jankovic, 2020) . For instance, functional chains represent functions and their logical and execution relationships (including information and energy flows for example). For these reasons, we rely specifically on SoS functional chains to build an SoS functional network.

Based on SoS functional chains, shown on the left of Figure 1, we define the SoS functions dependency matrix  $A = [a_{i,j}]_{i,j \in [1,n]}$ , where  $n$  is the number of considered SoS functions. An element  $a_{i,j}$  of matrix  $A$  is equal to 1 if function  $i$  requires the output of function  $j$  as its input and 0 otherwise. In other words,  $(a_{i,j} = 1)$  if the execution of function  $i$  depends on function  $j$ . Thus,  $A$  is a squared non-symmetric matrix. The graph  $G$  corresponding to the dependency matrix  $A$  is unidimensional, directed, and unweighted, as shown in the middle of Figure 1.  $G$  represents an SoS functional network, as shown in the right of Figure 1.

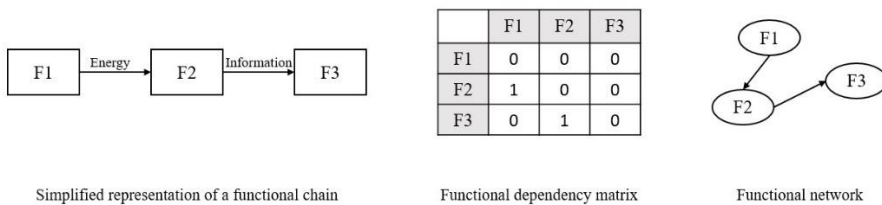


Figure 14. Building an SoS functional network from SoS functional chains

### 3.2 Network-based metrics to characterize SoS functional change and failure

In the following paragraphs, we detail the definitions of the local network-measures we considered for characterizing the tendency of functions to propagate change and failure and the impact a function's change or failure on the robustness of the network. We also propose an interpretation of such measures when looking at an SoS functional network.

- a) Local centrality measures: characterizing the tendency of functions to propagate change and failure

- The **degree centrality** simply counts the number of edges linked to a given node. In the context of a directed network, we differentiate in-degree centrality and out-degree centrality. In-degree centrality counts the number of edges incident to the node ( $nb\_in\_ed$ ) while out-degree count the number of edges linked to the node and pointing toward other nodes ( $nb\_out\_ed$ ). Equations (1) and (2) give the formula for normalized in- and out-degree centrality measures. In general, in/out-degree centrality measures the importance of a node looking at its direct connections. Considering the SoS functional network, a high out-degree centrality value indicates that the function is highly relied-upon (i.e., provides inputs to many other functions) and might cause cascading changes and failures. A high

in-degree centrality value reveals that a function highly relies on its neighbor functions and can be changed or can fail as soon as one of its neighbor functions undergoes a change or a failure.

$$DC_{in} = \frac{nb_{in\_ed}}{n - 1} \quad (1)$$

$$DC_{out} = \frac{nb_{out\_ed}}{n - 1} \quad (2)$$

Degree centrality considers direct links alike. Thus, it does not include the quality of the links or the cyclic dependency.

- The **eigenvector centrality** defines the node importance with respect to the importance of the nodes to which it is connected. The importance of node  $i$  ( $x_i$ ) can then be written as ( $\lambda$  being a constant):

$$x_i = \frac{1}{\lambda} \sum_{j \in \Gamma(i)} x_j \quad (3)$$

With  $\Gamma(i)$  the neighborhood of node  $i$ .

The SoS functional network being directed, we differentiate node influence (i.e., function influence) and node susceptibility as suggested by (Sarica & Luo, 2019). The influence of a node is quantified by the number of edges pointing out to other nodes that are themselves influential (i.e., a function is as influential as it is highly relied-upon by functions that are themselves influential). The susceptibility of a function is proportional to the number of susceptible functions on which it relies on. Influential functions tend to propagate changes and failures in the networks, while susceptible functions are susceptible to changes and failures that are carried to them.

Replacing the importance of a node  $i$  ( $x_i$ ) by its influence ( $p_i$ ) and its susceptibility ( $q_i$ ) in equation (3), it can be written in a matrix form as equation (4) and (5) for influence and susceptibility, respectively.

$$A^t \mathbf{p} = \lambda_p \mathbf{p} \quad (4)$$

$$A \mathbf{q} = \lambda_q \mathbf{q} \quad (5)$$

Where  $\mathbf{p}$  and  $\mathbf{q}$  are vectors of size  $n$  with  $\mathbf{p}[i] = p_i$  and  $\mathbf{q}[i] = q_i$  for  $i \in \llbracket 1, n \rrbracket$ .

According to the Perron–Frobenius theorem, Eq. (4) and (5) have unique solutions (with non–negative indices  $p_i$  and  $q_i$ ), which are the eigenvector corresponding to the largest eigenvalue  $\lambda_p$  and  $\lambda_q$  of  $A^t$  and  $A$  respectively. Both vectors  $\mathbf{p}$  and  $\mathbf{q}$  can be normalized as  $(\frac{n}{\sum_{i \in \llbracket 0, n \rrbracket} p_i} \mathbf{p})$  and  $(\frac{n}{\sum_{i \in \llbracket 0, n \rrbracket} q_i} \mathbf{q})$  (Sarica & Luo, 2019).

- b) Topological dysconnectivity and  $\Delta$ ASPL: characterizing the impact a function's change or failure on the robustness of the network

In a network, a “vulnerable” node is such that its removal disconnects a large portion of the network or increases the ASPL (Walsh et al., 2018). Thus a “vulnerable” node is a node

that diminishes the robustness of the network. In the context of SoS development, a change or a failure of a CS corresponds to the removal of a function (or multiple functions) or deterioration of functional dependencies. Such change or failure can jeopardize the overall functionality of the SoS. When a function is removed, its direct dependencies are also removed. The functional chain is consequently shredded and the overall SoS functionality may be lost. Similarly, when the functional dependencies are deteriorated, the functional chain and the overall SoS functionality may be degraded. Two options are considered to characterize the consequences of a change or a failure undergone by a specific function on the network. The first option is to consider the number of connected networks and the size of the largest connected network after its removal; The second option is to compute the  $\Delta ASPL$  after a deterioration of its direct functional dependencies materialize by a decrease in the weight of all edges linked to it from 1 to 0.5 (fault variable suggested by (Walsh et al., 2018)). Since, the aim is to assess the effect of the function deterioration on the overall robustness of the network, the network is considered undirected and incoming and outgoing edges are equivalent. Equation 6 reminds the ASPL formula ( $d_{i,j}$  being the shortest distance between node  $i$  and  $j$ ).

$$ASPL = \frac{1}{n^2} \sum_{i,j \in [1,n]} d_{i,j} \quad (6)$$

The higher the number of connected graphs generated by a function removal; the more effort can be expected to recover the overall functionality of an SoS. The more extensive the largest connected network, the more robust the network is to the function removal. Looking at the  $\Delta ASPL$ , higher values indicate that the function is more “vulnerable” compared to other nodes in the network. Thus, high  $\Delta ASPL$  indicates that a deterioration of a function’s dependencies (as a consequence of its change or failure) degrades the overall SoS functionality.

Both topological dysconnectivity and  $\Delta ASPL$  can be measured after the removal or deterioration of multiple functions, thus reflecting multiple changes or failures.

#### 4 Network-Base metrics comparison and implications on a use case

We compute the local network-metrics introduced in section 3 on a functional network corresponding to an SoS related to Electric Vehicles (EVs). A car manufacturer, an energy provider, and eventually several service providers collaborate to offer EV charging service (Plug & Charge) in the context of Electric Vehicle to Grid (V2G) services (Chazal, 2018). Such a service is an SoS as it requires independent systems (CSs) (e.g., EV, charging stations, electricity grids, and service provision servers) to function together to provide the EV charging service. Based on the operational and functional descriptions of the service provision modeled in SysML, we counted 8 CSs performing 22 high-level functions to provide the service. We construct the functional dependency matrix and the functional network of the SoS from the available functional chains (Figure 1). Table 1 presents the values of in- and out-degree centrality and eigenvector centrality measures (influence and susceptibility score). It also gives the number of connected networks and the size of the largest connected network after the function removal and the  $\Delta ASPL$  after the deterioration

of the function direct dependencies. In both Figure 2 and Table 2, CSs and CSs functions are identified by their Ids.

According to the results in Table 2, most functions have the same tendency to propagate changes and failures to their neighbors. The same tendency can be observed for functions to undergo a change or a failure when the neighbor functions change or fail (in-degree equals out-degree for functions 1,2,3,4,5,11,12,15,16,18, and 22). Function 6 highly relies on its neighbor functions and, as such, is highly susceptible to its neighbor functions changes and failures. On the contrary, function 10 presents a high out-degree centrality measure and can cause cascading changes and failures to its neighbors. Since the considered directed network (Figure 2) does not present cycles and has low connectivity, influence and susceptibility indicators do not allow to clearly differentiate functions (function 7 being the most influential and function 6 the most susceptible).

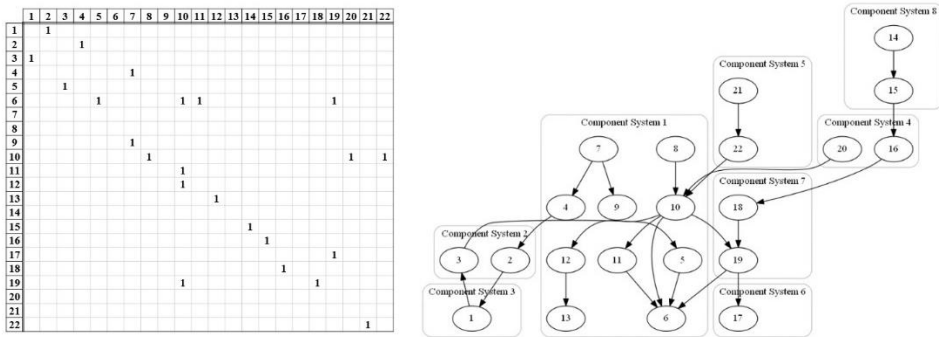


Figure 15. SoS functional dependency matrix and corresponding functional network

Besides, changes or failures of functions with high  $\Delta ASPL$  values (functions 10, 6, and 5, for example) are likely to degrade the SoS overall functionality. Functions 10 and 19 seem critical as their removal disconnects the network in 5 and 3 connected networks, respectively. Such disconnection might involve a significant effort to retrieve the overall SoS functionality. Both the removals of functions 6 and 7 disconnect the network into 2 connected networks. However, the largest connected networks caused by the removal of function 7 is larger than that caused by the removal of function 6 (20 and 14 nodes). This might indicate that the removal of function 6 has a larger impact on the overall network robustness than the removal of function 7.

Therefore, these measures allow the identification of functions critical to the propagation of change and failure or functions whose change or failure degrades the overall functionality of the SoS. During the SoS early design stages, such measures allow the SoS decision-makers to prioritize the redevelopment of critical functions or allocate them to CSs of which they control the management and operation. Furthermore, local network-metrics complemented by global-network metrics can be used to compare different SoS functional architectures and assess “what-if” scenarios. Thus, network-metrics can aid SoS decision-makers to identify which SoSs are worth getting involved in. During operation and runtime, these measures help track the critical functions as the system evolves.



Table 4: Network metrics measures for the considered EV related SoS functional network

Id	In-degree	Out-degree	Influence Indicator	Susceptibility Indicator	$\Delta$ ASPL	Nb of disconnected networks	Size of the largest connected network
1	0,048	0,048	0	0	0,340	2	17
2	0,048	0,048	0	0	0,279	2	18
3	0,048	0,048	0	0	0,392	2	16
4	0,048	0,048	0	0	0,210	2	19
5	0,048	0,048	0	0	0,435	2	15
6	0,190	0	0	22	0,496	2	14
7	0	0,095	22	0	0,132	2	20
8	0	0,048	0	0	0,045	1	21
9	0,048	0	0	0	0,045	1	21
10	0,143	0,190	0	0	0,517	5	15
11	0,048	0,048	0	0	0,045	1	21
12	0,048	0,048	0	0	0,132	2	20
13	0,048	0	0	0	0,045	1	21
14	0	0,048	0	0	0,045	1	21
15	0,048	0,048	0	0	0,132	2	20
16	0,048	0,048	0	0	0,210	2	19
17	0,048	0	0	0	0,045	1	21
18	0,048	0,048	0	0	0,279	2	18
19	0,095	0,095	0	0	0,409	3	16
20	0	0,048	0	0	0,045	1	21
21	0	0,048	0	0	0,045	1	21
22	0,048	0,048	0	0	0,132	2	20

## 5 Conclusion and future work

In this paper, we use an SoS functional network and measure degree and eigenvector centrality to characterize the influence or susceptibility of a function to change and failure propagation in its direct neighborhood and the network in general. We also assess the tendency of the SoS to maintain its functionality after a function removal or a function deterioration. Through an example on an EV related SoS, we show that such measures are complementary and can aid an SoS decision-maker during the SoS development or runtime. However, further industrial applications are to be analyzed to confirm the usefulness of network-metrics. Other network-metrics, including global metrics, can be used in the analysis of SoS evolution, including changes and failures. Furthermore, theoretical work is still to be done to better understand the possible uses of graph theory in SoSs, specifically in terms of results interpretation. This is all the more important as SoSs involve not only heterogeneous and independent systems and their functionalities but also independent stakeholders. Therefore, they can be modeled as multilayer networks.

## References

- Antul, L., Ricks, S., Mann, L., Cho, K., Cotter, M., Jacobs, R. B., ... Dahmann, J. (2017). Toward Scaling Model-Based Engineering for Systems of Systems. *2018 IEEE Aerospace Conference*, 1–9.
- Chai, C. L., Liu, X., Zhang, W. J., & Baber, Z. (2011). Application of social network theory to prioritizing Oil & Gas industries protection in a networked critical infrastructure system. *Journal of Loss Prevention in the Process Industries*, *24*(5), 688–694. <https://doi.org/10.1016/j.jlp.2011.05.011>
- Clarkson, P. J., Simons, C., & Eckert, C. (2004). Predicting change propagation in complex design. *Journal of Mechanical Design, Transactions of the ASME*, *126*(5), 788–797. <https://doi.org/10.1115/1.1765117>
- Fakhfakh, S., Hein, A. M., Chazal, Y., & Jankovic, M. (2020). A meta-model for product service systems of systems. In *Accepted in Design 2020*.
- Haley, B. M., Dong, A., & Tumer, I. Y. (2016). A Comparison of Network-Based Metrics of Behavioral Degradation in Complex Engineered Systems. *Journal of Mechanical Design, Transactions of the ASME*, *138*(12), 1–11. <https://doi.org/10.1115/1.4034402>
- Keating, C. B., & Katina, P. F. (2011). Systems of systems engineering: prospects and challenges for the emerging field. *International Journal of System of Systems Engineering*. <https://doi.org/10.1504/IJSSE.2011.040556>
- Li, Y., Wang, Z., Zhong, X., & Zou, F. (2018). Identification of influential function modules within complex products and systems based on weighted and directed complex networks. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-018-1396-9>
- Luo, J. (2015). A simulation-based method to evaluate the impact of product architecture on product evolvability. *A Simulation-Based Method to Evaluate the Impact of Product Architecture on Product Evolvability*. <https://doi.org/10.1007/s00163-015-0202-3>
- Luo, J. (2017). System-of-systems innovation : Proactive methods for conception and strategies for implementation. *Journal of Enterprise Transformation*, *0*(0), 1–25.
- Maier, M. W. (1996). Architecting Principles for Systems-of-Systems. *INCOSE International Symposium*, *6*(1), 565–573. <https://doi.org/10.1002/j.2334-5837.1996.tb02054.x>
- Mohsin, A., Janjua, N. K., Islam, S. M. S., Vicente, V., & Neto, G. (2019). A Taxonomy of Modeling Approaches for Systems-of-Systems Dynamic Architectures : Overview and Prospects.
- Papariotodimou, G., Duffy, A., Whitfield, R. I., Knight, P., & Robb, M. (2020). A network tool to analyse and improve robustness of system architectures. *Design Science*, (May), 1–40. <https://doi.org/10.1017/dsj.2020.6>
- Petitdemange, F., Borne, I., & Buisson, J. (2018). Modeling system of systems configurations. In *13th Annual Conference on System of Systems Engineering (SoSE)* (pp. 392–399). <https://doi.org/10.1109/SYSoSE.2012.6384152>
- Sarica, S., & Luo, J. (2019). An Infinite Regress Model of Design Change Propagation in Complex Systems. *IEEE Systems Journal*, *13*(4), 1–9. <https://doi.org/10.1109/jsyst.2019.2899988>
- Sosa, M., Mihm, J., & Browning, T. (2011). Degree Distribution and Quality in Complex Engineered Systems. *Journal of Mechanical Design*, *133*(10), 101008. <https://doi.org/10.1115/1.4004973>
- Uday, P., & Marais, K. (2015). Designing Resilient Systems-of-Systems: A Survey of Metrics, Methods, and Challenges. *Systems Engineering*, *18*(5), 491–510. <https://doi.org/10.1002/sys.21325>
- Walsh, H. S., Dong, A., & Tumer, I. Y. (2018). The role of bridging nodes in behavioral network models of complex engineered systems. *Design Science*, *4*, 1–28. <https://doi.org/10.1017/dsj.2017.31>
- Walsh, H. S., Dong, A., & Tumer, I. Y. (2019). An Analysis of Modularity as a Design Rule Using Network Theory. *Journal of Mechanical Design, Transactions of the ASME*, *141*(3), 1–10.