On the Concepts of Capability and Constituent System Independence in Systems-of-Systems

Jakob Axelsson[1, 2]
[1] Mälardalen University
Västerås, Sweden
jakob.axelsson@mdu.se

Pontus Svenson[2]
[2] RISE Research Institutes of Sweden
Kista, Sweden
pontus.svenson@ri.se

***Abstract*—Systems-of-systems are designed to provide a capability that their constituent systems cannot achieve individually. A key property is that the constituent systems have some degree of operational and managerial independence. The concepts of capability and independence are thus central to the field of systems-of-systems. Yet the contemporary literature and standards only give vague definitions of these terms. This vagueness is a barrier to progress in the field, and this paper aims at contributing with a more detailed conceptualization. It describes a system capability as a state-transforming process that uses certain resources. Independence means that the system has a choice about when and how its capabilities should be activated. This requires that the system is an intelligent agent with a notion of utility, a perception of the world around it, and a decision-making capability. When given a mission, the system can complete that mission by activating appropriate combinations of capabilities. A system-of-systems can decompose its mission into parts that correspond to the capabilities of various constituent systems.**

Keywords—system-of-systems, capability, independence, mission, constellation.

I. Introduction

A system-of-systems (SoS) can informally be seen as a set of collaborating independent systems. From its roots in the defense sector, the concept is nowadays being used in a broad range of industrial and societal domains, such as transportation, energy, manufacturing, and crisis management. This increased usage is an effect of widespread digitalization. Consequently, more general and efficient methods for SoS engineering (SoSE) need to be developed, which in turn demands a clear and precise terminology for reasoning about SoS.

A. Current System-of-Systems Concepts

A pioneering contribution to SoSE was Maier's characterization of SoS, which included: operational independence of constituent systems (CS); managerial independence of CS; emergent behavior; evolutionary development; and geographical distribution [1]. Of these, the first two which emphasize CS independence are the core, whereas the others can be derived from general system properties.

Maier's notion of CS independence remains central also in contemporary accounts of SoSE. Yet, there are no clear descriptions of what independence means, which makes it a poor foundation for constructing useful SoS models.

A more recent step in maturing SoSE is a set of standards that use the following definitions of key terms [2]:

- *SoS:* "set of systems or system elements that interact to provide a unique *capability* that none of the CS can accomplish on its own."
- *CS:* "*independent* system that forms part of an SoS."
- *Capability*: "measure of capacity and the ability of an entity (system, person or organization) to achieve its objectives."

This research was funded by Vinnova (Sweden's innovation agency), grant no. 2019-05100 and the Knowledge Foundation, grant no. 20200230.

There are, as will be seen in the next section, several different interpretations of the term capability. We will argue that the one used in the SoS standards has a serious flaw.

B. Purpose and Contribution

The purpose of this paper is to improve our understanding of SoSE by providing improved definitions of key concepts such as capability, independence, and missions, and their interrelations. This can be seen as the embryo of an ontology that will improve our ability to reason about SoS and communicate among stakeholders. This is particularly important in the context of SoSE since this reasoning and communication does not only occur among engineers during design time, but due to the continuous SoS evolution, it will also take place during operation and throughout the lifecycle.

One aspect of the SoS dynamics is the formation of constellations of CS [3], to address current operational needs and missions by instantiating SoS capabilities. Finding the appropriate constellations requires the SoS to reason about the mix of capabilities provided by different CS. The efficiency of this reasoning can be improved by having a digital information model representing capabilities and similar concepts, which requires a more precise definition of terms. The information model can also be used as the basis for applying model-based techniques to SoSE.

This paper's main contribution is a conceptualization of capabilities. This turns out to also mandate the clarification of other concepts, such as CS independence. With this as a foundation, it becomes possible to move one step closer to the vision of digital engineering of SoS.

C. Overview of the Paper

The remainder of the paper is structured as follows: In the next section, some related work is introduced, with a focus on capability engineering. Then, in Section III, we introduce a new conceptualization of capability. Section IV explains how a CS can be described as independent, based on its ability to choose when and how to activate its capabilities. Section V shows how a joint SoS capability can be created by a constellation of CS, and how the capabilities can be used to carry out a mission. Section VI discusses the proposed concepts from a modeling and analysis perspective, and in the final section, the paper is summarized together with suggestions for future research.

II. Related Work

We will now describe some previous research related to capability definitions, and capability and mission engineering.

A. Capability Definitions

Henshaw et al. [4] investigate the meaning of "capability" in the literature through a structured method and identify eight different world views that can be mapped to different stakeholder categories. Different definitions of capability in the risk management field are reviewed in [5], which highlights that the term often involves access to resources and sometimes also relates to objectives.

A three-layer structure for describing defense capabilities is suggested in [6]. The first layer consists of platforms (aircraft, ships, etc.) and facilities (training, personnel, infrastructure, doctrines, organization, information, etc.) The second layer consists of functions that should be achieved, such as command and control, airlift, or deep strike. The third layer identifies effects that should be obtained. Another three-dimensional set of views is proposed by [7], including systems, lifecycle, and operational capability elements.

B. Capability Engineering

The term capability engineering has been given increasing attention over the last decade, primarily in crisis management and defense, and with a goal to improve terminology. Yue and Henshaw [8] attempt to provide some clarity into capability development, by specifying terminology and mapping the main stakeholders and organizational entities involved.

Several researchers have tried to devise frameworks for developing capabilities, such as [9] which describes an architecture for analysis of capability needs, assessment of capability options, and making choices about requirements. The analysis starts with scenarios and challenges, from which critical capability components are identified. This leads to a mission-system analysis where different combinations of options are compared, as a basis for system acquisition.

Pei et al. [10] investigate a method for assessing capability gaps in an SoS, i.e., the degree to which a designated SoS action plan cannot be implemented. They model the situation at hand through a conditional evidential network where belief functions are provided using estimates of requirements satisfaction probabilities. These are used to identify the weak points among the CS, to which more resources should be diverted to close the gap.

INCOSE UK has produced a short guide to capability systems engineering [11]. They stress that capabilities are enduring, even though the realization may change, and that people are involved in creating them. The components include people, processes, equipment, and organization. The human aspects are often neglected and have been called "the elephant in the room" of SoSE [12]. Therefore, the guide recommends using elements of Soft Systems Methodology (SSM) [13] as part of capability engineering.

Much of the previous references relate to the defense domain, but capabilities are also used in the business IT context [14]. The paper emphasizes capability maps of the entire enterprise. It notes that capabilities should be named using nouns, and often nominalized verbs are used.

C. Mission Engineering

A systematic literature mapping of how missions should be described was performed by [15], resulting in a conceptual description of the key elements. The main elements are tasks, triggers, and constraints. Based on this conceptual model, a language for modeling missions was proposed by [16]. It is based on the KAOS formalism originally developed for requirements modeling. The notation is exemplified using a flood monitoring example.

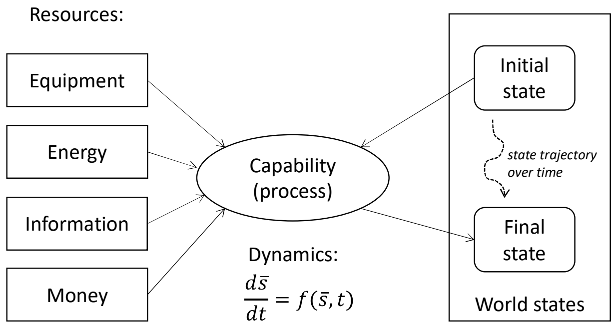

Fig. 1. Key elements of the capability concept.

In [17], an ontology for describing SoS missions is proposed, and they discuss principles for how the mission may be decomposed and allocated to CS. A mission model is analyzed to detect common mistakes before execution. The approach is exemplified through an air defense application.

D. Summary of Contributions to Body of Knowledge

This paper extends previous work by providing a set of abstract concepts that are essential when reasoning about capabilities. This is useful as the basis for applying model-based techniques for capability and mission engineering.

III. Capability Concept

We will now introduce our view on a suitable conceptualization of a system's capability. First, we will clarify which of the different interpretations of capability we subscribe to. Then, the nature of capability as a process will be described, followed by an account of the resources used and the dynamics created. Finally, a small example is introduced to illustrate the various aspects. The key elements of the concept are illustrated in Fig. 1.

A. Deficiencies with Current SoSE Definition of Capability

As pointed out by [5], capability definitions in the literature differ on whether they include objectives as part of the capability, or not. As mentioned in Section I, the current SoSE standards define capability as a "measure of capacity and the ability of an entity (...) to achieve its objectives."

We will now argue that including objectives as part of capability is a serious flaw. First of all, it is different from the everyday use of the word, which in most dictionaries is defined as "the ability to do something" [18], or similarly. It also deviates from classic work in systems thinking, such as Beer, which defines capabilities as "what we *could* be doing ... with existing resources, under existing constraints" [19] (p. 163).

The more fundamental issue is that varying the objective does not change what the system could do and should thus not be seen as changing its capability. As an example, consider a system consisting of a car and a driver. The driver uses the car to go back and forth to work, which would be the objective. The system has the capability of transporting a person. Assume now that the driver suddenly gets the need to transport some smaller goods. This is something the system has obviously all along been capable of, but with the standard's definition, it does not have that capability until the driver decides to change objective.

We will return to this small example later in the paper. It should be noted that the car with a driver is not a good example of an SoS but it can be seen as a CS of a larger transportation SoS. The SoS aspects will be touched upon in Section V.C below.

The standard's definition puts an unfortunate focus on a stated current objective, which tempts the SoS engineers to provide a minimal solution to that particular objective. This will reduce the flexibility to deal with changing circumstances.

In this paper, we will follow Beer's (see above) and others' interpretations, and intuitively define capability as "something a system could do." We regard objectives as something separate, which should be seen as part of missions (see Section V).

B. *Capability as a State Transforming Process*

As pointed out above, people are an essential component of capability, and in line with the recommendation in [11], we let the soft systems methodology (SSM) inspire our definition of the capability concept.

SSM [13] is a method that aims at intervening in a situation that is perceived as problematic. The situation is modeled through purposeful activities, which are state-transforming processes, carried out by actors who hold certain world views.

If an actor can perform an activity, it must have the corresponding capability, and hence we can regard a capability as being fundamentally a state-transforming process as well. This means that the capability can be activated from a certain set of world states, and when activated it will, after some time, bring the world to a state which meets certain conditions.

C. *Resources*

As seen in Section II above, capabilities require resources of different kinds. These resources are required for the transformation carried out by the capability process. If the resources are not available, the capability cannot be fully used.

In general, resources can be classified along two orthogonal dichotomies, as shown in Table 1. The first relates to whether the resource has a relevant physical shape (tangible) or not (intangible). The other relates to whether the resource is destroyed when it is used (consumable) or whether it remains and can later be used again (durable).

Note that even though a durable resource remains after completing the capability, it could be exclusive to a limited number of users at a time. For example, a piece of equipment could only be used to deliver one capability at a time. Once that capability has been fulfilled, the equipment is available again for another task.

A common reason for collaborating in an SoS is for CS to get access to resources that are controlled by other CS, in particular information resources.

One may note that the usage of resources by the capability can also be seen as a world state change, but we choose to highlight the resources since they are often easy to identify in domain problems.

TABLE I. RESOURCE TYPES

	Tangible	**Intangible**
Consumable	Energy	Money
Durable	Equipment People	Information Process descriptions Organizations Mission descriptions

D. *Capability Parameters and Process Dynamics*

Capabilities are things that the system could do. However, often the same capability can be achieved in somewhat different ways by the system. It is a process that transforms some initial state into another state. This final state should meet certain criteria, but there may be multiple such states. Also, since the capability is a process, it takes some time, during which the world proceeds through certain intermediate states. There can be different pathways through those intermediate states, that can take different amounts of time, and different amounts of resources may be consumed as a consequence. The capability thus has, in some sense, parameters allowing it to be activated in different ways, which creates a different final result and with different dynamics.

As pointed out in [11], the abstract capabilities often remain the same in an SoS throughout its evolution, but the implementation can be altered. This also means that the parameters and dynamics can change to provide greater utility.

E. *Example*

To illustrate the elements of capability, the example introduced in Section III.A will now be continued. Consider a car with its driver. This system has a transportation capability, which allows the system to be displaced from point A to point B. In the process, it uses resources such as energy (the fuel), equipment (the road), information (a map), and possibly money (to pay road tolls). The capability can be activated in different ways, allowing it to travel to any destination which is connected through a road network. The process dynamics can also be altered, by driving at different speeds. This can influence the resources consumed, such as the amount of fuel. Some resources need to be provided through the capabilities of other systems, such as the fuel infrastructure.

IV. CONSTITUENT SYSTEM INDEPENDENCE

After having discussed the nature of a capability, and concluded that the capability of a system is a state-transforming process that uses some resources, we will now turn the discussion to the independence of CS.

In our view, the fundamental aspect of CS independence is that it can decide how and when to activate its capabilities. This decision-making element requires that we regard the CS as an intelligent agent with some perception of the world and some notion of value, which it uses to choose alternative courses of action. In the remainder of this section, we will discuss those aspects in more detail. Fig. 2 illustrates the main concepts introduced.

A. *Perception and the World Model*

An intelligent agent must have some perception of the surrounding world, so it is an open system with information interfaces. This perception allows it to maintain a world model, which is an internal information resource. The world model contains both its understanding of the current state of

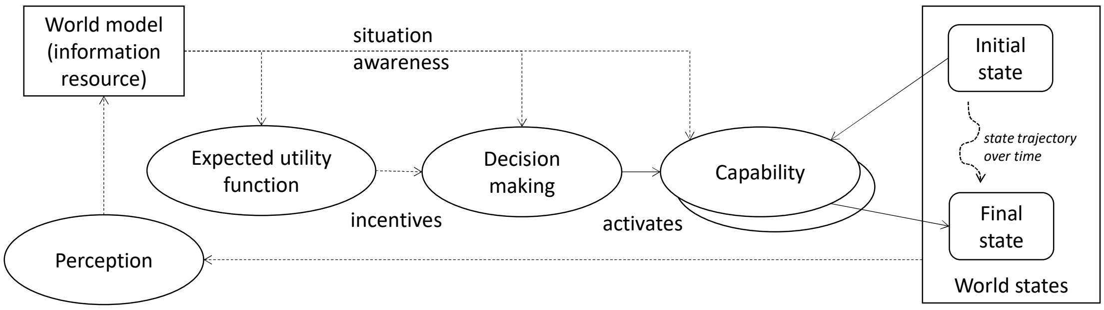

Fig. 2. Elements of a decision-making agent representing a CS.

the world, at a certain abstraction level, but also abstract notions about the mechanics of the world, which enables it to predict how the world will evolve. This creates a situation awareness [20], which it uses both operatively when a certain capability is active, but also for planning how to use its capabilities in the future.

B. Expected Utility Function

The decision on how the CS should activate its capabilities is based on what effects will be achieved, and the CS will strive to get the best possible effects. This requires some notion of value, which we will capture as an expected utility function. It is a mapping from the set of states of the world that it can perceive, to some notion of utility. Since only perceivable states are considered, many different world states are equivalent from the agent's point of view. The utility needs to be at least a partial ordering, meaning that some states are more valuable than others. For some other pairs of states, the agent can be indifferent regarding their relative utility, or they may not be easily comparable. (We are not concerned here with how the agent calculates its utility, but only that it must have some notion of value that gives a partial order of states.)

The agent does not have full control over the changing state of the world, but other agents are acting in it as well, and natural processes have their course. Therefore, the agent needs to consider an expected utility, involving some uncertainty, rather than absolute utility.

C. Decision Making

The decision the independent CS needs to make is what capabilities to activate and in what way (i.e., which parameters to provide). Apart from maximizing its expected perceived utility, it will also need to consider certain constraints, such as access to resources.

Most CS will have multiple capabilities, and some of them may be possible to activate at the same time, but others could require the same resources. As a simple example, a person may have the capability of running and the capability of load carrying. However, it may not be capable of both simultaneously.

The CS needs to consider its utility over time, which means that it can sometimes choose to, e.g., take a short-term cost to achieve a greater utility later on. A good example of this is when a CS makes some investments to prepare for joining an SoS, from which it will later benefit.

Another aspect relates to the activities of other agents and their effects on expected utility. The CS may try to anticipate these actions, through its world model, and also communicate to align its actions with other CS, to avoid the game-theoretic dilemma that may otherwise occur and that could lead to a lower utility for everyone [21].

D. Example

Continuing the transportation example, the person that wishes to travel from A to B usually has a limited perception of the current road conditions for alternative routes, including congestion. The expected utility involves many factors which are hard to compare, and which are valued differently by different individuals, such as travel time, cost, comfort, and CO_2 emissions. In the decision, the actions of others may be anticipated, such as choosing to travel at a certain time to avoid expected congestion. When making the decision, uncertainties also need to be considered and depending on the nature of the travel, the person can be more or less risk-averse.

V. Constellations and Missions

In the previous two sections, we have described the nature and elements of capabilities, and how CS can decide on the usage of their capabilities through independent decision making. Now, we will see how several CS can collaborate in an SoS to create new capabilities and use these to carry out missions.

A. Constellations

An SoS creates new capabilities through CS collaboration. As for a CS, these SoS capabilities can be activated in different ways, and the activation requires that a particular subset of CS collaboratively activate certain of their own capabilities. This subset is called a constellation [3].

In terms of our view of capabilities as state-transforming processes that use resources, an SoS capability would also transform the world state from an initial to a final state. During that process, the different CS would perform their state transformations. One example of this is seen in Fig. 3, where the two CS in the constellation work in sequence. The first CS transforms the world into some intermediate state, and the second takes it from there to the final state. In other situations, the CS may work in parallel on orthogonal aspects of the world state, or one CS may be providing resources to another which performs the actual state change intended by the SoS capability.

B. Missions

A mission would represent something a system should do. Sometimes, the mission is assigned to the system by some other system, and sometimes, an independent agent can assign a mission to itself as a result of planning how to achieve some longer-term utility. Regardless of which, the mission is essentially an information entity.

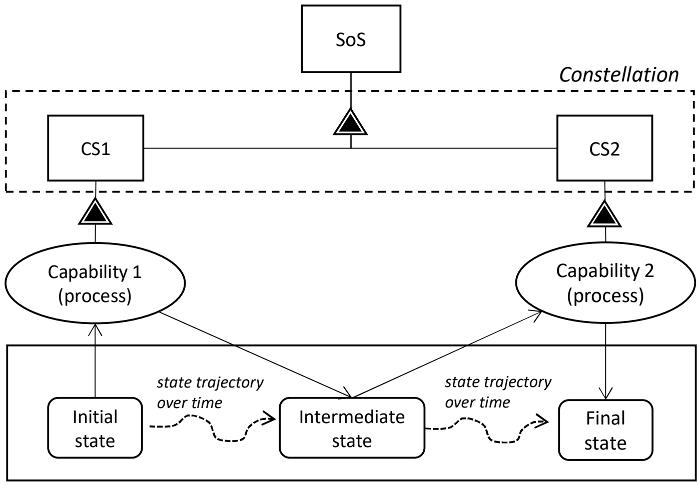

Fig. 3. A constellation of CS collaborating to provide an SoS capability through sequential activation of individual capabilities.

The mission may be broken down hierarchically into tasks to be carried out in sequence or parallel. To achieve an SoS mission, this decomposition is necessary to assign tasks to the CS that need to collaborate to carry out the overall SoS mission.

For the SoS mission to be successful, each mission task that is given to a CS must correspond to the capabilities the CS possesses, and the resources needed must be available. Also, the expected utility of the independent CS must be sufficient, so that it actually chooses to activate that capability.

C. *Example*

Returning once more to the transportation example, the person wishing to travel could have other options than driving. For instance, there could be public transportation available for a part of the route, such as driving to the railway station and then taking the train. This extends the example to an SoS situation. It is similar to the situation depicted in Fig. 3, with an intermediate state achieved by one CS and the final state by another. Another example is if two persons form a constellation to share a ride.

VI. Discussion

Due to the complexity of the situations dealt with, SoSE fundamentally needs to use model-based practices. A challenge is to find the appropriate constructs to use for efficient modeling, as well as supporting analysis and decision making. The purpose of this paper is to suggest a step in that direction. In this section, some of the modeling and analysis aspects of the proposed conceptualization are discussed.

A. *Modeling Considerations*

The purposes of modeling include aligning different stakeholders' views of the system-of-interest by making them explicit and providing support for analysis to support decision making during the design and operations phases. Key qualities of a modeling approach include efficiency, i.e., human resources needed for modeling, and effectiveness, i.e., how well the models support communication and analysis.

In this work, we have tried to address those qualities by defining a set of interrelated concepts focusing on key aspects of SoS, namely capabilities, CS independence, and to some extent missions. These concepts can be seen as a starting point in developing an ontology or meta-model, that guides analysts in the process of capturing stakeholder world views and making them explicit.

We have found that the elements used, such as capabilities, resources, perception, world models, utility functions, etc., are relevant to include in an ontology. Although they are abstract, they are still sufficiently close to the language used by stakeholders to be useful tools in collecting and structuring information from them.

B. *Class vs. Instance Information*

One distinction which is neglected in much SoS literature, and also in the previous parts of this paper, is the distinction between "class" and "instance". However, this distinction is commonly used in modeling languages, and some clarification is needed.

Most of the concepts introduced in this paper and others can be used to denote both class and instance. In the case of the SoS as a whole, there can be many SoS that share common features, such as various regional transportation systems. Each of them can be seen as a concrete instance. When designing the SoS, it makes sense to model it on a class level, defining abstract roles, responsibilities, and types of the CS participating, rather than mentioning concrete CS instances. Also, even if there is only a singleton SoS with similar characteristics, that SoS changes over time as CS join and leave it.

The same applies to constellations. There would be a class level constellation, which defines what types of CS could join, and how responsibilities would be distributed. Once a concrete set of CS starts to collaborate, they form a constellation instance.

In Section III.D, it was discussed how capabilities can be activated in various ways, leading to different state transition dynamics. This indicates that it makes sense to view capabilities mainly as a class-level construct during design time, and the activation is achieved by instantiating that class. The same applies to missions, which also benefit from being defined abstractly, and parameterized for instantiation.

C. *Analyses Supported by Models*

The conceptualization described in the paper has been driven partly by particular analysis needs, which are critical to SoS. This includes business aspects in commercial domains, where it is not only sufficient to have an overall SoS profitability but where there is also a need to ensure that each CS gains, since they would otherwise not choose to join or remain in the SoS. These aspects and the ensuing dynamics can be expressed through the expected utility functions, and how resources are transferred within the SoS.

Another area is risk and safety analysis, where the problem becomes how to maximize the chances that the SoS capabilities are available even in non-favorable situations. By modeling the CS capabilities, resources, and decision-making, new types of risks can be identified, making mitigation possible.

D. Operative Exchange of Capability Related Information

In Section V.B, it was explained how an SoS can break down a mission into tasks, which match the capabilities, resources, and incentives of CS. To do this in practice, the agents in the SoS that plan missions must have information that describes the capabilities of different CS, as well as their notion of utility and the resources available. All these aspects may change over time, and hence it is necessary for an SoS to operationally exchange such information between CS. The proposed concepts can also be used for such models [22].

VII. Conclusions and Future Research

This paper was based on the observation that some of the key concepts in contemporary SoSE have only been shallowly defined. This includes the terms capability and CS independence. By providing clearer and more elaborated descriptions of these concepts, it is anticipated that more effective SoS modeling can be achieved, leading to better analysis and decisions during design and operation.

The capability concept proposed here dismisses the idea that an objective should be part of the definition. Instead, it views capabilities as state-transforming processes that use different kinds of resources. The independence means that a CS can choose itself how to activate its capabilities. This requires it to be an intelligent agent with perception, a world model, a notion of utility, and a decision-making ability. In an SoS, constellations of CS are formed to provide emergent capabilities, and an SoS mission can be decomposed into missions carried out by CS with the appropriate capabilities.

The work in this paper has evolved over several years and has been driven by needs in application-oriented projects in various domains, including transportation (truck platooning [23], urban mobility [24]), construction (quarries and road works [25][26]), and the process industry. We are currently expanding this range of applications to also include, e.g., societal crisis management and defense. As part of this, we will continue to evolve the ontology and provide further analysis methods based on it.

References

[1] M. W. Maier, “Architecting Principles for Systems-of-Systems,” INCOSE Int. Symp., vol. 6, no. 1, pp. 565–573, Jul. 1996.

[2] ISO/IEC/IEEE, “Standard 21840 Systems and software engineering - Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of System of Systems Engineering,” 2019.

[3] J. Axelsson, “A Refined Terminology on System-of-Systems Substructure and Constituent System States,” in *IEEE Systems of Systems Conference*, pp. 31–36, 2019.

[4] M. Henshaw et al., “Capability Engineering–An Analysis of Perspectives”, In INCOSE International Symposium, 2011.

[5] H. Lindbom, H. Tehler, K. Eriksson, and T. Aven, “The capability concept - On how to define and describe capability in relation to risk, vulnerability and resilience,” Reliab. Eng. Syst. Saf., vol. 135, pp. 45–54, 2015.

[6] C. Kerr, R. Phaal, and D. Probert, D, “A strategic capabilities-based representation of the future British armed forces,” Intl. Journal of Intelligent Defence Support Systems, 1(1), 27-42, 2008.

[7] A. Harding, J. Mollett, and M. Touchin, “A structured approach to planning and managing systems engineering capability evolution in a complex through-life business space,” In 7th Annual Conference on Systems Engineering Research, 2009.

[8] Y. Yue and M. Henshaw, “An holistic view of UK military capability development,” Defense & Security Analysis, 25(1), 53-67, 2009.

[9] P. Davis, “Analytic architecture for capabilities-based planning, mission-system analysis, and transformation,” RAND National Defense Research Inst., Santa Monica, 2002.

[10] D. Pei, et al., “Prioritization Assessment for Capability Gaps in Weapon System of Systems Based on the Conditional Evidential Network,” Applied Sciences, 8(2), 2018.

[11] INCOSE UK, “Capability Systems Engineering Guide,” 2014.

[12] M. Henshaw et al., “The Systems of Systems Engineering Strategic Research Agenda”, Report of T-AREA-SoS project, 2013.

[13] P. Checkland and J. Poulter, “Learning for Action”, Wiley 2006.

[14] Ulrich, W. and Rosen, M, “The Business Capability Map: The ‘Rosetta Stone’ of Business/IT Alignment”, Enterprise Architecture, vol. 14, No. 2, Cutter Consortium, 2011.

[15] E. Silva et al., “On the characterization of missions of systems-of-systems”, Proc. European Conf. on Software Architecture, 2014.

[16] E. Silva, T. Batista, and F. Oquendo, “A mission-oriented approach for designing system-of-systems,” Proc. 10th System of Systems Engineering Conference, pp. 346-351, 2015.

[17] W. Zhu, H. He and Z. Wang, "Ontology-Based Mission Modeling and Analysis for System of Systems," IEEE International Conference on Internet of Things, Exeter, 2017, pp. 538-544.

[18] Cambridge dictionary [online]. https://dictionary.cambridge.org/dictionary/english/capability.

[19] S. Beer, “The Brain of the Firm,” 2nd Ed., Wiley 1981.

[20] P. Svenson and J. Axelsson, “Situation Awareness and Decision Making for Constituent Systems,” Proc. 15th Intl. Conf. on System of Systems Engineering, pp. 361-366, 2020.

[21] J. Axelsson, “Game theory applications in systems-of-systems engineering: A literature review and synthesis,” in 17th Annual Conf. on Systems Engineering Research, pp. 154–165, 2019.

[22] J. Axelsson, “Experiences of Using Linked Data and Ontologies for Operational Data Sharing in Systems-of-Systems,” in *IEEE Systems Conference*, 2019, pp. 396–403.

[23] J. Axelsson, “An initial analysis of operational emergent properties in a platooning system-of-systems,” in *12th Annual IEEE International Systems Conference*, 2018, pp. 1–8.

[24] J. Axelsson and S. Nylander, “An Analysis of Systems-of-Systems Opportunities and Challenges Related to Mobility in Smart Cities,” in *13th Annual Conference on System of Systems Engineering*, Jun. 2018, pp. 132–137.

[25] J. Axelsson, J. Fröberg, and P. Eriksson, “Towards a System-of-Systems for Improved Road Construction Efficiency Using Lean and Industry 4.0,” in *13th Annual Conference on System of Systems Engineering*, Jun. 2018, pp. 576–582.

[26] J. Axelsson, J. Fröberg, and P. Eriksson, “Architecting systems-of-systems and their constituents: A case study applying Industry 4.0 in the construction domain,” *Systems Engineering*, vol. 22, no. 6, pp. 455–470, Nov. 2019.