Towards a Core Ontology for Missions and Capabilities in Systems of Systems

Joyce Martin, Jakob Axelsson, Jan Carlson, Jagadish Suryadevara
School of Innovation Design and Engineering
Mälardalens University
Västerås, Sweden
(joyce.martin, jakob.axelsson, jan.carlson, jagadish.suryadevara)@mdu.se

***Abstract*—This study presents a core ontology for missions and capabilities in systems of systems. The aim of the study is to create artifacts that facilitate precise understanding of fundamental concepts of SoS. An ontological approach proposes and develops taxonomic homogeneity and structural knowledge of SoS. The ontology development process involved workshop sessions with industry experts and meta modelling of the different concepts and their relationships. The ontology includes thirteen concepts of types physical things, people, information and mixed. These concepts are defined and their relations are briefly described. The developed ontology is further illustrated using a wildfire scenario case study.**

***Index Terms*—core ontology, system of systems, mission, capability**

I. Introduction

The System of Systems (SoS) concept is gaining considerable importance to date. Although it is inspired by the defense sector, it is widely applied in different fields and industries including: healthcare, energy, telecommunications and military [1]. The subject of SoS brings complexity which primarily stems from the autonomy and heterogeneity of its constituent systems (CS) and their interrelations. This complexity is further enhanced by the engineering and management efforts put forth in the design and operation of the SoS. With such complexity, managing SoS uncertainties and risks take the center stage in SoS design and operations processes.

SoS exploit capabilities to accomplish missions through systems and interfaces [2]. Therefore, the usefulness of an SoS is a function of how well its capabilities are configured to achieve its missions. This must be achieved while taking into consideration different factors such as mission context, heterogeneity of the CS and stakeholders. This means taking into account both internal and external factors affecting the operational states of the SoS, and hence its performance.

The complexity of the SoS domain prompts a structured approach to distinguish and manage different levels of abstraction involved. Our approach looks at capabilities and missions as the cornerstone of SoS due to the recursive nature of the association of the two concepts. An ontological approach to SoS missions and capabilities will provide good guidance towards understanding fundamental constructs and their interconnection towards the emergent behavior of the SoS.

This research was funded by KKS, grant no. 2020-0230.

A. Ontologies

Ontology as a discipline has its foundation in philosophy, and it has been widely applied in other areas including information systems [3]. Guarino et al. [4] show the evolution of the definition of ontology from traditional philosophy to computation. Philosophy defines ontology as *"a field that studies concepts, such as existence, being, becoming, and reality"*, while the computational view defines ontology as *"a formal, explicit specification of a shared conceptualization"* [4]. Ontology is therefore a broad field where knowledge is centralized in well-defined concepts. Xiao-yong et al. [5] outline the categorization of ontology elements into five types, i.e. *classes, relationship, function, axioms* and *instance*. Therefore, an ontological approach to understanding concepts studies the nature and relations surrounding the concepts.

There are different categories of ontologies. The widely used categorization of ontology is discussed by [6], [7]. This defines three broad categories, i.e., *foundational*, *core* and *domain* ontologies. These categories correspond to generic reference across fields, structured reference in one field, and specific knowledge representation for a particular domain respectively [7]. For consistency this study makes reference to these three types of ontologies with a particular focus on the core ontology. An ontological approach to studying a domain enables separation and re-use of knowledge as well as facilitating communications and interoperability [8], [9].

B. Contribution

The main contribution of this paper is the creation of the core ontology for SoS. SoS leverage the interoperability of independent systems, which makes complexity an inevitable factor in design and development of SoS. With this point of view, understanding of basic and minimal [10] concepts that build up the complexity of SoS is important. These fundamentals are provided by a core ontology [10].

A core ontology is application agnostic i.e. *"The structured knowledge of the core ontologies is clearly separated from the domain-specific knowledge"* [7] and provides a structure that can be instantiated through different domain ontologies of the specific field, SoS in this case.

C. Overview of the Paper

The remainder of this paper is structured as follows: Section II briefly outlines the methodology used in developing the ontology. Section III is on related work giving an overview of ontologies for SoS. Section IV shows the structure of the core ontology and taxonomy of its concepts. Section V illustrates the core ontology with a wildfire scenario. Section VI discusses the results and Section VII concludes and outlines future work.

II. Methodology

This study employed design science methods [11] which included:

1) Iterative ontology development process.
2) Workshops with stakeholders.
3) Case study analysis of existing ontologies.
4) Illustration of the ontology in a typical operational scenario.

The involved stakeholders are industry practitioners in systems and software engineering with expertise in the construction (Volvo) and defense (SAAB) industries.

Although literature points out the lack of an all-in-one methodology to developing ontologies, a lot of the development methodologies are motivated by experience in certain contexts including enterprise architecture and agile engineering [12], [13], [14]. This study uses highlights of these identified steps but limited to the purpose of a core ontology. The involved steps include: identification of scope and purpose of the ontology, re-use of existing knowledge and development of a conceptual model.

III. Related Work

This section summarizes related work in SoS related to ontological approaches. These studies are with regard to metamodelling, propositions, interoperability, mission and reviews of various issues of interest.

A. SoS Ontology Metamodels and Propositions

A number of studies have talked about ontology meta modelling in SoS. Baek et al. [15] developed a metamodel for representing SoS where their focus was on requirements that differentiate CS and SoS entities in terms of their goals, infrastructure, environment, organization, requirements, service and capability, and their corresponding environmental factors. Knöös et al. [16] gave a holistic view of an SoS through an ontological and reasoning approach with an objective of exploring the SoS design space. This provides very useful ideas for both CS and SoS developers and users who are also the main stakeholders of this study.

Yan et al. [17] identified four elementary propositions in constructing an SoS ontology: the objective existence of SoS, influencing internal and external factors, independent interrelationships of these factors, the abstract nature of SoS, and high dependence on expert knowledge. This highlights many concerns which we may group into two: abstraction and interoperability issues. Abstraction elaborates on the need for expert knowledge in understanding SoS. In developing our core ontology, this was achieved using domain knowledge experts workshop meetings. Nilsson et al. [18] explored the practical implication of SoS ontology between two companies, Volvo and Uber ATG. The study explored the correspondence attached to each ontology term in a practical operational case applicable in both companies. The results showed how use of an ontology can enable collaboration in an operational context, further motivating this study.

A review study by Abdalla et. al. [19] on knowledge representation approaches in SoS showed the dominance of SoS ontology studies compared to taxonomy, thesaurus and vocabulary. These knowledge representation studies are motivated by the need for terminology standardization, support for SoS integration and management as well as support for SoSE activities [19]. However, these studies are more focussed on specific themes as opposed to general view of SoS. There are lessons we can learn from these theme-specific studies, lessons that can help us conceptualize and characterize SoS. The next section gives an overview of different constructs related to SoS missions with profound ontological implications.

B. SoS Mission Modelling

With mission context as the foundation of SoS configuration, Silva et. al. [20] characterized missions to be of predominantly three types: achievement, maintenance and avoidance missions. They outlined missions as composed of eight elements, and used five elements to create a conceptual mission model: *task, trigger, priority, parameter, constraints*, where other elements served as more supporting tasks:*executor, final condition and relationship* [21]. They went further to model SoS missions through a semi-formal language mKAOS [22]. mKAOS is a derivative of KAOS, a goal-oriented requirement engineering methodology [21]. mKAOS collates models that link different aspects of SoS. These are: mission model which shows the global SoS goal, intermediary goals and individual CS goals, responsibility model linking CS and environmental agents, object model for data and events, operational and communication capability models and emergent behavior model. However, mKAOS mission modelling is a design time activity, and not executable, and moreover not supportive of runtime evolution, dynamism, inter-dependance and sequencing of events [22]. In SoS context this is a drawback. The study went a step further and proposed a formal language to improve on the possibilities of verification of the mission models.

Silva et. al. [23] further extended their work on Software Intensive Systems of Systems (SiSoS) and mapped the mKAOS models to architecture descriptions to take into account both structural and behavioural viewpoints [24]. This work refined mKAOS mission models to provide architecture descriptions through a formal Architecture Description Language (ADL) [23]. An ADL takes into account structure, behavior, interactions, adaptations, properties, constraints and quality attributes [23], and these are important in SoS.

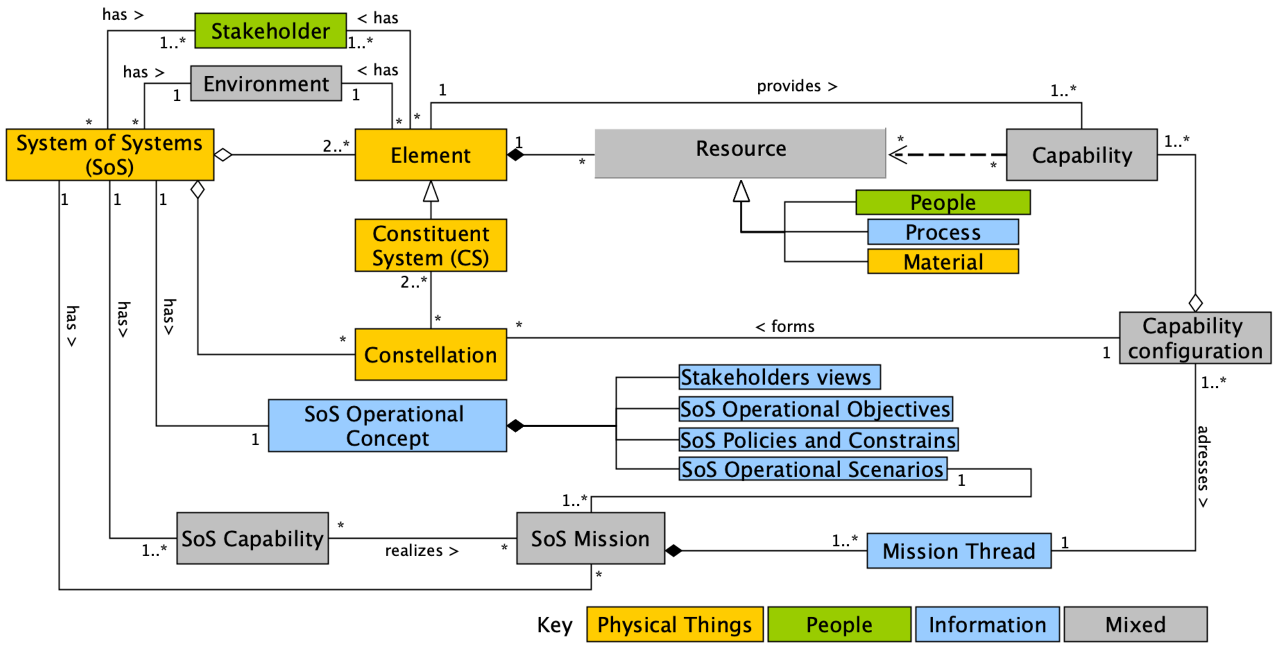

Fig. 1. The Core Ontology

C. *SoS Ontology and Interoperability*

Axelsson [25] looked at how the Levels of Conceptual Interoperability model (LCIM) and semantic web technology can help in ontological understanding of SoS interoperability issues. Benali et al. [26] worked on SoS interoperability and developed a context-based ontology, aimed at creating a guideline for SoS interoperability concerns. In their ontology, the authors looked at interoperability problems in SoS and different characteristics of SoS and their influence on interoperability. The ontology describes SoS as contextually made up of its relational as well as intrinsic nature. This approach relates to the separation of concerns used by Baek et al. [15], but in a more extensive way as the approach by Benali et al. is hinged on a foundational ontology.

On the other hand whilst studying interoperability, John et al. [27] looked at three levels of ontology layers: domain, mission and instance levels. In their work, mission ontology correlates mission, capability and activity, whereas instance ontology correlates system and function. The study further bridged these two with a service layer, an approach that adds modularity in the event a new system is added to support the SoS missions.

Zhu et al. [28] used an ontological approach to decompose SoS missions. This study defined decomposition patterns, principles and algorithms and further formalized the mission concept. This showed: decomposition, supporting and conflicting nature of missions, and henceforth the importance of mission models in supporting the SoS requirement model [28]. Fakhfakh et al. [29] developed a meta model for Product Service Systems of Systems (PSSoS). Their study links the service and the value provided to the customer with complex relationships in SoS to bridge the gap between PSS and SoS concepts [29].

Two systematic reviews by Lana et al. [1] and Martin et al. [30] show a multitude of SoS development initiatives which include: processes, frameworks, approaches, methods and models. These show the characteristics, properties, constraints, and opportunities associated with different SoS aspects. These studies highly complement our approach and understanding as we developed a core ontology which is described in the next section.

IV. The Core Ontology

This section describes the developed core ontology which is depicted in Fig 1. The ontology identifies 13 main concepts and their relationships, including some composition elements and instances. To address its concepts, the ontology adapts to existing taxonomies of definitions. These definitions and relations are as described in each concept.

A. *System of Systems*

"A set of systems or system elements that interact to provide a unique capability that none of the constituent systems can accomplish on its own" [31]. An SoS has stakeholders and exists in an environment. It is made up of two or more elements that can be combined into constellations of capabilities. An SoS has a defined operational concept, designed to fulfill SoS missions through unique SoS capabilities collated from the capabilities of the SoS elements.

B. *Stakeholders*

"A part that has interest in, or is affected by, outcomes or intermediate effects generated or influenced by the SoS"

definition adapted from [32]. An SoS has stakeholders and so do its elements. The intersection of these stakeholders produces diverse views that drive the operational concept of the SoS.

C. Environment

"A definition of the environmental factors in which something exists or functions" [32]. Both the SoS and its elements exist in an environment. This implies shared concerns on these factors and their intersection.

D. Element

"One of the parts which make up the whole thing" [33]. An SoS is made up of elements, which include two or more constituent systems and other elements introduced by the environment or the stakeholders in addressing a respective mission context.

E. Constituent System (CS)

"Independent system that forms part of a system of systems" [34]. CS form the main parts of the SoS elements that provide required capabilities towards fulfilling an SoS Mission.

F. Resource

"A composite structure representing the physical and human resources (and their interactions), assembled to meet a capability", modified from [32]. Resources originate from the elements of SoS and provide a baseline for possible capabilities. This ontology categorizes resource into three main types: *people, process,* and *material*.

G. Capability

"The ability to achieve a desired effect under specified (performance) standards and conditions through combinations of ways and means (activities and resources) to perform a set of activities" [35]. Capabilities use resources, are organized in capability configurations, and are aimed at specific activities. Resources, be it *people, process,* and *material* may stand-alone or be combined entities providing certain capabilities.

H. Constellation

"A subset of CS that have formed links with other CS, allowing them to exchange information, therefore playing an important role as the provider of a particular SoS capability" [36]. A constellation is an instance of two or more CS designated to achieve a specific mission thread. It is the constellation that realizes a capability configuration for corresponding CS taking specific roles towards providing SoS Capability.

I. Capability Configuration

"A composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to provide a capability", [32]. In an SoS this composite structure represents the combination of different capabilities required by a specific SoS mission thread. Capability configurations guide the usefulness of capabilities in different mission contexts.

J. SoS Operational Concept

"A user-oriented document that describes a system's operational characteristics from the end user's viewpoint" [37]. This concept is about continuously managing the usefulness of the independence of CS and their roles towards the SoS. It describes the different stakeholder views, operational objectives, policies, constraints, and scenarios. These translate to achievable goals, variability guide and consequently strategic visions of the SoS.

K. SoS Mission

"A set of objectives and goals to be achieved in a specific operational environment" [38]. A mission is accomplished through steps described as mission threads; these correspond to specific functions achievable through CS capabilities. The specific operational environment corresponds to the SoS operational scenario and therefore defines the contextual nature of missions. SoS missions are contextual in nature, as they exist in environments with varying factors. SoS missions stem from the operational concept, which is expected to contain a variability guide that is useful to address the different missions contexts.

L. Mission Thread

"A sequence of end-to-end activities and events, given as a series of steps, that accomplish the execution of one or more capabilities that the SoS supports" [39]. Missions can be viewed from a high-level and also be decomposed into sub-missions and tasks. These smaller missions and tasks are achieved through one or multiple mission threads. These correspond to functions achievable through CS capabilities. Mission threads guide SoS configuration preparedness.

M. SoS Capability

"The ability of the SoS to exhibit unique behaviour(s) through capability configurations of its associated CS", modified from [35]. Literature references this as achieving the emergent behaviour of the SoS [36], [40]. This capability results from the achievement of SoS mission objectives and goals.

V. Wildfire Case Study Illustration

This section illustrates how the different concepts in the ontology can be used in practise through a case study on a wildfire scenario. This shows the functionality and applicability of the conceptualization employed in the core ontology in an operational scenario. A study on wildfire management and safety [41] elaborated the decentralized Swedish approach to wildfire management. The study surveyed incident commanders' (IC) resource dimensioning, tactics and perception of wildfire scenarios, and noted the dependency of mission effectiveness on the experience of the IC. This means varying experiences of IC's perception and consequently implementations largely affect mission success rates [41]. With such variations, a wildfire scenario is a good example to demonstrate this core conceptualization.

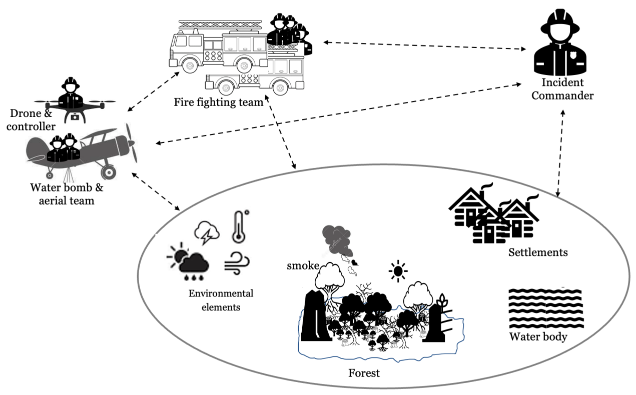

Fig. 2. Wildfire Scenario

A. Scenario Description

The suggested scenario is visualized in Figure 2. The emphasis is on a fire detection mission. The flow of events starts when a possible wildfire is reported by civilians as they see smoke coming from the densely forested area about 20 km from their settlements. The municipality commissions response efforts which are led by an IC. Considering the danger posed by the fire regarding human settlements and forest reserves, police forces or even military response may be deployed depending on the severity of the incident.

Based on a study that interviewed ICs [41], a typical initial deployment in fire fighting by Swedish authorities includes, at minimum 2 ground fire-fighting units, i.e., 2 officers and 8 fire-fighters, from two different stations, each with one engine (3 m^3, one pump and 500 m hose), one tanker (8-10 m^3) and with drone capabilities. Fig. 3 summarizes the various concepts and their interconnections, following the proposed ontology. We see the heterogeneity of the stakeholders and their influence in the operational concept. From a practical perspective, we look at the operational concept as a combination of decisions and priorities in response to the scenario, which can be different in different scenarios. We see how the main characteristics (CHAR) of each concept play a role in influencing the decision and subsequently the overall performance of the SoS.

To simplify this interconnection, the scenario is described in three layers: generation, use and sustenance, and decision layers. These layers extend the military capability management framework defined by [42] which includes capability generation, sustainment and employment.

- *Generation layer:* maps resources to generate capabilities, with the main concerns hinged on resource types and their availability as well as the effectivess of the available capability. This is a demonstration of how CS manage their independence yet contribute to the SoS, in this layer each CS aligns its opportunities and constraints in participating in the SoS.
- *Usage and sustainment layer:* creates a routine to use and sustain capabilities, i.e. by mapping capabilities to missions and tasks through configurations. It considers stakeholder, mission context, performance measures, thread dependencies, and the efficiency of the possible constellation towards the desired SoS capability.
- *Decision layer:* filters the characteristics and influences of the different concepts towards achieving the desired outcome. This step determines the actual System of Interest (SoI) that addresses the mission threads taking into account the mission context.

These layers show the different dimensional concerns of SoS which are hinged on; independent systems, functional dependability and heterogeneity in involved systems. The scenario illustrates the transformation and operational states of the SoS. Transformation shows the integration of the CS through constellation of configurations and the operational state shows the prospective outcomes in relation to the generation, usage, sustainment, and decision making in the use of capabilities.

B. Ontology Utilization

The utilization of the ontology shows the kind of issues the core ontology can be used for. As a basis of discussion, these are herein explained in connection with the wildfire scenario:

- The definition of the elements of the SoS and their associated resources and capabilities. The wildfire scenario identified six constituent systems and one mediator.
- The multi-stakeholder and heterogeneity concerns which result from the diversity of the CS and their usefulness to the SoS. The scenario links local and national agencies, contractor, civilians and teams with varying expertise and interests.
- The complexity of mission definition, where sub-missions are identified and may have overlapping dependencies and parallelism in e.g. terrain assessment (TA) and fire behavior analysis (FBA) threads in the scenario. These open the scenario to identification of the most used CS, direct better scheduling, and allow SoS users and developers to avoid common pittfalls such as redundant threads.

Overall this is aimed at making users and developers think of the rigor of analyzing each of the concepts. This is a step towards having the right capability configurations and efficient constellations that use available resources well and prioritize dependencies. Such an analytical view sees to it that the right requirements are taken into account in fulfilling the SoS capability through well directed mapping of elements, resources, capabilities, configurations and constellations. To document it all, the operational concept shows that the SoS needs and expectations are met. In the scenario this is the decision support system that weighs the characteristics (CHAR) of different concepts and use expert opinions to choose the trade-offs and the way forward.

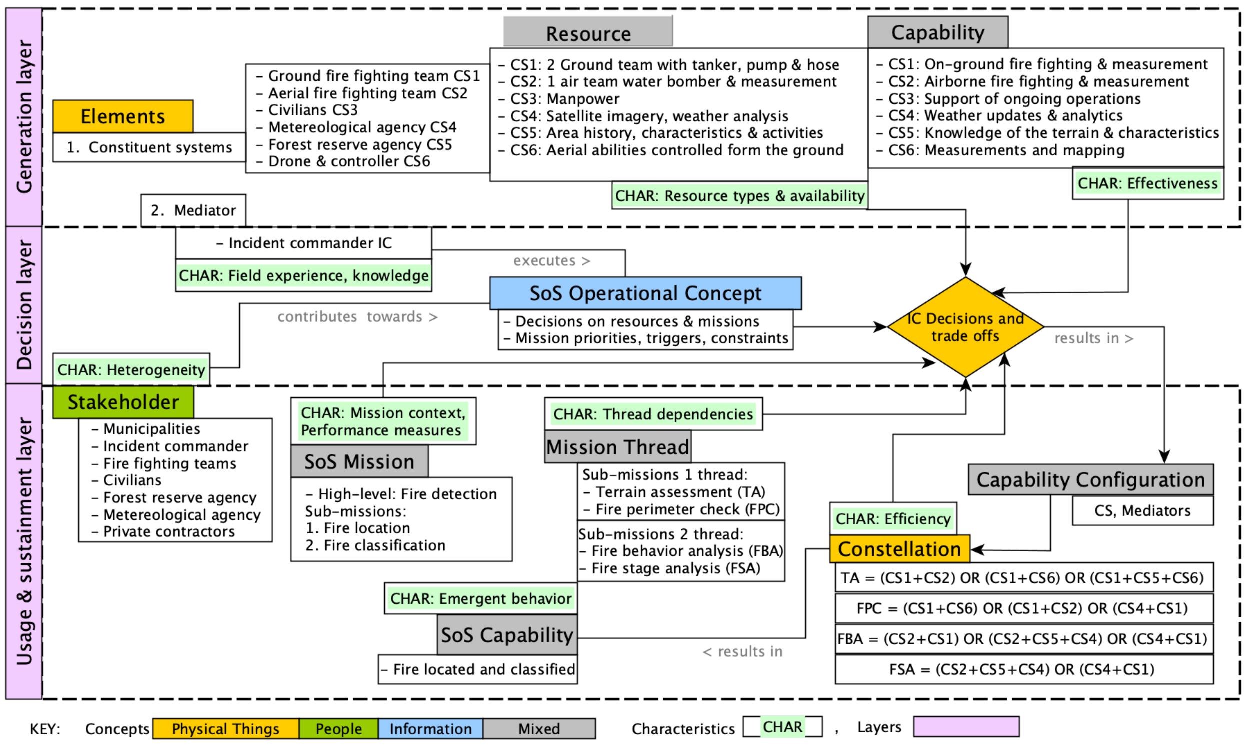
Generation layer
Decision layer
Usage & sustainment layer
Elements
1. Constituent systems
2. Mediator
– Ground fire fighting team CS1
– Aerial fire fighting team CS2
– Civilians CS3
– Metereological agency CS4
– Forest reserve agency CS5
– Drone & controller CS6
Resource
– CS1: 2 Ground team with tanker, pump & hose
– CS2: 1 air team water bomber & measurement
– CS3: Manpower
– CS4: Satellite imagery, weather analysis
– CS5: Area history, characteristics & activities
– CS6: Aerial abilities controlled form the ground
CHAR: Resource types & availability
Capability
– CS1: On–ground fire fighting & measurement
– CS2: Airborne fire fighting & measurement
– CS3: Support of ongoing operations
– CS4: Weather updates & analytics
– CS5: Knowledge of the terrain & characteristics
– CS6: Measurements and mapping
CHAR: Effectiveness
– Incident commander IC
CHAR: Field experience, knowledge
executes >
SoS Operational Concept
– Decisions on resources & missions
– Mission priorities, triggers, constraints
contributes towards >
CHAR: Heterogeneity
IC Decisions and trade offs
results in >
Stakeholder
– Municipalities
– Incident commander
– Fire fighting teams
– Civilians
– Forest reserve agency
– Metereological agency
– Private contractors
CHAR: Mission context, Performance measures
SoS Mission
– High–level: Fire detection
Sub–missions:
1. Fire location
2. Fire classification
CHAR: Thread dependencies
Mission Thread
Sub–missions 1 thread:
– Terrain assessment (TA)
– Fire perimeter check (FPC)
Sub–missions 2 thread:
– Fire behavior analysis (FBA)
– Fire stage analysis (FSA)
Capability Configuration
CS, Mediators
CHAR: Efficiency
Constellation
TA = (CS1+CS2) OR (CS1+CS6) OR (CS1+CS5+CS6)
FPC = (CS1+CS6) OR (CS1+CS2) OR (CS4+CS1)
FBA = (CS2+CS1) OR (CS2+CS5+CS4) OR (CS4+CS1)
FSA = (CS2+CS5+CS4) OR (CS4+CS1)
CHAR: Emergent behavior
SoS Capability
– Fire located and classified
< results in
KEY: Concepts
Physical Things
People
Information
Mixed
Characteristics
CHAR
Layers

Fig. 3. Wildfire Scenario Illustration

VI. Discussion

What the core ontology tells us with regard to development and use of SoS is not only the fundamental concepts, but also the need for purpose and structure. These determine why and how CS form the desired emergent behavior of the SoS. SoS are purposeful entities whose complexity adjusts with the number of involved elements and their corresponding process. With this in mind, an SoS can implement various SoI depending on their capability richness. Therefore, SoS developers and users can exploit the range of capabilities to develop the emergent behavior, and preferably have a flexible module-like architecture that enables capability add-ons.

Purpose is addressed by a pre-conceived interaction of functional and non-functional variables. Functional variables are the capabilities that become useful in a constellation. Non-functional variables demonstrate the quality attributes of the mission threads towards the overall mission i.e. measurable variables that define the mission objectives and acceptable achievements.

Structures let us think of how the evolutionary and adaptive nature of SoS modifies the relationships between these concepts, and give an opportunity to add clarity when thinking of domain specific instances of the concepts. Both SoS and CS lifecycles are inherently different, yet one may have profound influence on the other. This implies a continued state of change and adaptation which is reflected in architectural as well as operational states of the SoS.

Several question then arise: is the list of minimal concepts captured in the core ontology?. Are the right issues addressed, i.e. can a CS developer look at the ontology and find meaningful concepts on how his CS can contribute in an SoS scenario?. Can an SoS administrator arrange the available capabilities to efficiently create the desired SoS emergent behaviour?. Answering these questions is subject to continued iteration of the ontology to a conceptualized shared understanding of SoS that can provide a proof of concept of this ontology in the SoS research and practitioners communities.

The discussion on precedence with regard to capability configuration raised different alternatives:

- Configurations are created in response to a mission thread and then SoS capabilities are derived from the configurations.
- SoS capabilities are defined first, then configurations are defined that provide those capabilities and when a mission thread is defined it can be associated with a suitable configuration.

This raised concerns on what type of association is fit among concepts surrounding capability configuration. This corresponds to how one looks at an SoS, either as a loosely coupled, a configured or a strategically planned system. Since the ontology is aimed as a guide for SoS developers and users, we choose to see the SoS as a planned system making the SoS capability/ emergent behavior a precedence.

VII. Conclusions and future work

The complexity of SoS highlights the need for abstraction techniques and ontological approaches. The proposed ontology aims to address communication and ease in system engineering. These can be seen to have been addressed in the proposed ontology. This work has the potential to correlate the various methodological initiatives in SoS research.

The continuation of this work points towards enhancing ontological interoperability by correlating the core ontology with a foundation ontology. This will make us rethink the concepts at a much broader perspective, to create an artifact that is in alignment and re-usable in a larger research domain. This will be followed by development of domain ontology and further verification of the core and domain ontologies.

References

[1] C. A. Lana, N. M. Souza, M. E. Delamaro, E. Y. Nakagawa, F. Oquendo, and J. C. Maldonado, "Systems-of-systems development: Initiatives, trends, and challenges," in *2016 CLEI*, pp. 1–12, IEEE, 2016.

[2] J. Dahmann, G. Rebovich, M. McEvilley, and G. Turner, "Security engineering in a system of systems environment," in *2013 IEEE Int. SysCon*, pp. 364–369.

[3] N. Guarino and P. Giaretta, "Ontologies and knowledge bases," *Towards very large knowledge bases*, pp. 1–2, 1995.

[4] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?," *Handbook on ontologies, Springer*, pp. 1–17, 2009.

[5] X.-y. Du, M. Li, S. Wang, *et al.*, "A survey on ontology learning research.," *Ruan Jian Xue Bao(Journal of Software)*, vol. 17, pp. 1837–1847, 2006.

[6] A. Gangemi, F. Fisseha, J. Keizer, J. Lehmann, A. Liang, I. Pettman, M. Sini, and M. Taconet, "A core ontology of fishery and its use in the fishery ontology service project," *EKAW04 Workshop, UK*, 2004.

[7] A. Scherp, C. Saathoff, T. Franz, and S. Staab, "Designing core ontologies," *Applied Ontology*, vol. 6, no. 3, pp. 177–221, 2011.

[8] T. R. Gruber, "The role of common ontology in achieving sharable, reusable knowledge bases," *Kr*, vol. 91, pp. 601–602, 1991.

[9] M. Obitko, V. Snasel, J. Smid, and V. Snasel, "Ontology design with formal concept analysis.," in *CLA*, vol. 128, pp. 1377–1390, 2004.

[10] DBpedia, "Core ontology." Available at https://dbpedia.org/page/Core$_ontology(2023/03/05)$.

[11] A. Dresch, D. P. Lacerda, J. A. V. Antunes Jr, A. Dresch, D. P. Lacerda, and J. A. V. Antunes, *Design science research.* Springer, 2015.

[12] N. F. Noy, D. L. McGuinness, *et al.*, "Ontology development 101: A guide to creating your first ontology," 2001.

[13] D. Jones, T. Bench-Capon, and P. Visser, "Methodologies for ontology development," 1998.

[14] S. Peroni, "A simplified agile methodology for ontology development," in *OWL: Experiences and Directions–Reasoner Evaluation: 13th Int. Workshop*, pp. 55–69, Springer, 2017.

[15] Y.-M. Baek, J. Song, Y.-J. Shin, S. Park, and D.-H. Bae, "A meta-model for representing system-of-systems ontologies," in *Proceedings of the 6th Int. Workshop on Software Engineering for Systems-of-Systems*, pp. 1–7, 2018.

[16] L. Knöös Franzén, I. Staack, P. Krus, C. Jouannet, and K. Amadori, "Ontology-represented design space processing," in *AIAA AVIATION 2021 FORUM*, 2021.

[17] H. Yan, Z. Jing, Y. Li-qun, L. Ze-min, and T. Li-jian, "Based on ontology methodology to model and evaluate system of systems (sos)," in *2014 9th Int. Conf. SOSE*, pp. 101–106, IEEE, 2014.

[18] R. Nilsson, S. Viswanathan, A. Mason, P. Jurland, P. Durgempudi, and J. Fischer, "Systems of systems ontology in practice," in *INCOSE Int. Symposium*, vol. 30, pp. 1383–1397, Wiley Online Library, 2020.

[19] G. Abdalla, C. D. N. Damasceno, M. Guessi, F. Oquendo, and E. Y. Nakagawa, "A systematic literature review on knowledge representation approaches for systems-of-systems," in *2015 IX Brazilian Symposium on Components, Architectures and Reuse Software*, pp. 70–79, IEEE.

[20] E. Silva, E. Cavalcante, T. Batista, F. Oquendo, F. C. Delicato, and P. F. Pires, "On the characterization of missions of systems-of-systems," in *Proceedings of the 2014 European Conference on Software Architecture Workshops*, pp. 1–8, 2014.

[21] E. Silva, T. Batista, and F. Oquendo, "A mission-oriented approach for designing system-of-systems," in *2015 IEEE 10th system of systems engineering conference (SoSE)*, pp. 346–351.

[22] E. Silva and T. Batista, "Formal modeling systems-of-systems missions with mkaos," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pp. 1674–1679, 2018.

[23] E. Silva, E. Cavalcante, and T. Batista, "Refining missions to architectures in software-intensive systems-of-systems," in *2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS)*, pp. 2–8.

[24] E. Silva, E. Cavalcante, T. Batista, and F. Oquendo, "Bridging missions and architecture in software-intensive systems-of-systems," in *2016 IEEE 21st International Conference on Engineering of Complex Computer Systems (ICECCS)*, pp. 201–206.

[25] J. Axelsson, "Achieving system-of-systems interoperability levels using linked data and ontologies," in *INCOSE Int. Symposium*, vol. 30, pp. 651–665, Wiley Online Library, 2020.

[26] H. Benali, N. B. Ben Saoud, and M. Ben Ahmed, "Context-based ontology to describe system-of-systems interoperability," in *2014 IEEE/ACS 11th AICCSA*, pp. 64–71, 2014.

[27] J. S. Osmundson, T. V. Huynh, and P. Shaw, "Developing ontologies for interoperability of systems of systems," in *Conference on Systems Engineering Research*, 2006.

[28] W. Zhu, H. He, and Z. Wang, "Ontology-based mission modeling and analysis for system of systems," in *2017 IEEE Int. Conference on Internet of Things (iThings)*, pp. 538–544, 2017.

[29] S. Fakhfakh, A. Hein, M. Jankovic, and Y. Chazal, "A meta-model for product service systems of systems," in *Proceedings of the Design Society: DESIGN Conference*, vol. 1, pp. 1235–1244, Cambridge University Press, 2020.

[30] J. Martin, J. Axelsson, J. Carlson, and J. Suryavedara, "The capability concept in the context of systems of systems: A systematic literature review," in *2022 IEEE ISSE*, pp. 1–8, 2022.

[31] 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.

[32] OMG, "Unified architecture framework modeling language (uafml) version 1.2," 2022.

[33] HarperCollins, "Collins dictionary - dictionary definitions." Available at https://www.collinsdictionary.com/dictionary/english/element (2023/03/05).

[34] ISO/IEC/IEEE, "Standard 15288:2015, systems and software engineering - system life cycle processes iso/iec/ieee 15288 in the context of system of systems engineering," 2015.

[35] DoD, "Department of defense architecture framework, version 2.02," 2009.

[36] J. Axelsson, "A refined terminology on system-of-systems substructure and constituent system states," in *2019 14th Annual Conf. SoSE*, pp. 31–36, 2019.

[37] IEEE, "IEEE guide for information technology–system definition–concept of operations (conops) document.," *IEEE Std 1362-1998*, pp. 1–24, 1998.

[38] OUSD(R&E), "Mission engineering guide," tech. rep., Approved for public, unlimited distribution, 2020.

[39] M. Gagliardi, W. Wood, and T. Morrow, "Introduction to the mission thread workshop," tech. rep., Software Engineering Institute, CMU, 2013.

[40] T. J. Inocêncio, G. R. Gonzales, E. Cavalcante, and F. E. Horita, "Emergent behavior in system-of-systems: A systematic mapping study," in *Brazilian Symposium on Software Engineering*, pp. 140–149, 2019.

[41] J. Sjöström, A. Granström, and L. Vylund, "Perception of wildfire behaviour and fire suppression tactics among swedish incident commanders," in *Advances in Forest Fire Research*, pp. 1733–1739, Coimbra University Press, 2022.

[42] G. Antunes and J. Borbinha, "Capabilities in systems engineering: an overview," in *Exploring Services Science: 4th Int. Conf., IESS, Portugal,*, pp. 29–42, Springer, 2013.