Obstacles of System-of-Systems

2022 IEEE International Symposium on Systems Engineering (ISSE) Proceedings

Tekinerdogan, Bedir

https://doi.org/10.1109/ISSE54508.2022.10005491

Obstacles of System-of-Systems

Bedir Tekinerdogan
Information Technology, Wageningen University & Research
Wageningen, The Netherlands
bedir.tekinerdogan@wur.nl

***Abstract*— System of systems (SoS) is applied for integrating various existing or newly developed systems to create value that cannot be obtained from the separate substituent single systems. Although there seem to be clear benefits to adopting SoS, integrating the independent substituent systems is not trivial. The existing literature has addressed several obstacles related to SoSs, but these are fragmented over different studies. This paper identifies and describes the obstacles of SoS and adopts a more comprehensive approach. The obstacles have been identified through both a domain analysis process and the experiences of industrial projects from different application domains.**

Keywords—system of systems, obstacles, case study research, domain analysis

I. INTRODUCTION

A system of systems (SoSs) is a configuration of systems that results when independent systems are integrated into a larger system that delivers unique capabilities [1][3]. SoSs have been largely applied in the defense domain, but obviously, SoSs are also apparent in many diverse application domains. In multiple different application domains, SoS is now applied for integrating various systems to create value that cannot be obtained from the separate substituent single systems. Although there seem to be clear benefits to adopting SoS, integrating the various substituent elements to create additional value is not trivial. Various obstacles have been reported in the literature, but these are fragmented over different studies. This paper identifies and describes the obstacles of SoS and aims to provide a more comprehensive perspective. The obstacles have been identified through both a domain analysis process and the experiences of running industrial projects. The industrial projects were carried out in different domains, including physical protection SoS, Smart Metering and Smart Grid SoS, Smart Farming SoS, C4ISR SoS, Healthcare SoS, Enterprise SoS, and Space SoS. The domain analysis process has resulted in the identification of primary studies related to the obstacles of SoS. The results of both sources have been synthesized and categorized. As a result, various obstacles have been identified, which are explained using the adopted case studies.

The results of this study can be beneficial for both researchers and practitioners. For researchers, the study outlines the key obstacles and thus paves the way and provides the direction for further research. For practitioners, the study helps to anticipate the possible obstacles when dealing with the design or analysis of SoSs.

The remainder of the paper is organized as follows. First, in section 2, we briefly describe the adopted research methodology. Then, section 3 presents the domain analysis process and the adopted case studies. Next, section 4 describes the identified obstacles, and finally, section 5 concludes the paper.

II. RESEARCH METHODOLOGY

Fig. 1 shows the adopted method for identifying the obstacles of SoS.

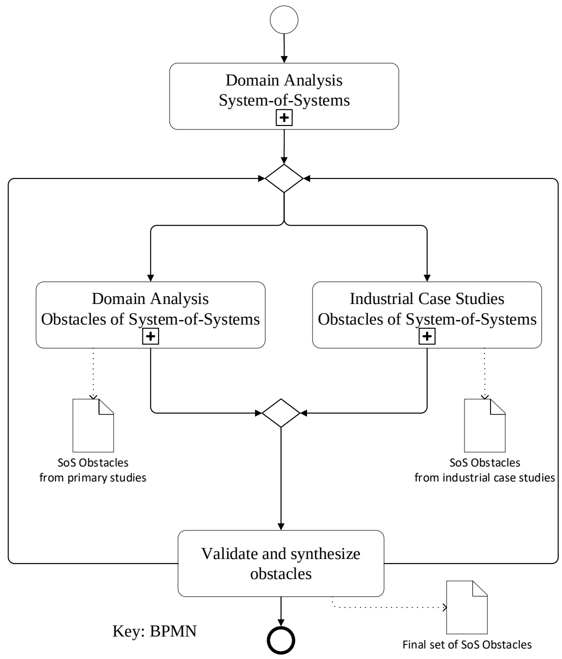

Fig. 1. Adopted research method

In the first activity of the method, we apply a domain analysis approach to model the key concepts related to SoSs. We have reported on this in our earlier publication [6]. This step is followed by a domain analysis process to the obstacles of SoS. This activity focuses explicitly on the reported obstacles in primary studies in the literature. The term *domain* refers to an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area. Domain analysis is defined as a systematic process for analyzing and modeling a domain [2]. Domain analysis consists of two sub-activities, including domain scoping and domain modeling. Domain scoping includes the definition of the domain of interest and selects the primary studies that will be used to extract the information necessary to answer the indicated research question. For this study, the domain scope includes the studies in SoS that describe the challenges. We exclude other primary studies and select a sufficient number of relevant studies. Based on the selected primary studies, the domain modeling is started, which results in a domain model. The domain model can be in different forms, including a natural text explaining the domain, a glossary that defines the key concepts, a conceptual diagram that shows the relations among the identified concepts, or a feature diagram used to describe the common variant features of the domain. In this study, we have chosen to describe the identified problems in natural text in table format.

Obstacles are also identified from industrial case studies in which the author was a primary researcher and or consultant during the last 15 years. The last activity validates and synthesizes the identified obstacles from previous domain analysis and case study activities. Multiple iterations might be applied between the last and earlier activities if needed. In the following sections, we will discuss these four activities in detail.

III. DOMAIN ANALYSIS AND INDUSTRIAL CASE STUDIES

A. *Domain Analysis*

Fig. 2 shows the adopted domain analysis process for modeling obstacles of SoSs. The general domain analysis process defines two distinct steps, including domain scoping and domain modeling. In the domain scoping process, the relevant primary studies are selected that can provide information about obstacles of SoSs. In the domain modeling process, relevant information from the selected primary studies is collected, a commonality and variability analysis is performed on the identified findings, the obstacles are described, and finally, these are evaluated and synthesized in a final set of obstacles.

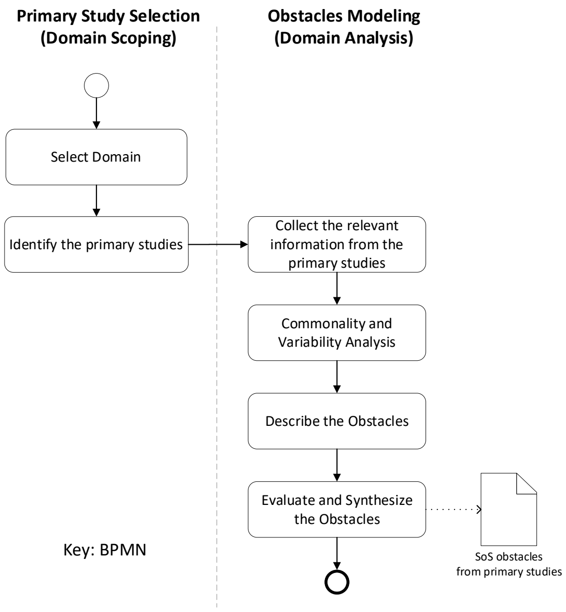

Fig. 2. Adopted domain analysis process for modeling obstacles of SoS

B. *Industrial Case Studies*

Fig. 3 shows the process for identifying obstacles derived from industrial projects. In total, seven industrial projects were considered that were performed over the last 15 years. The author was a researcher/consultant/system architect in these projects and had direct contact and collaboration with the industrial stakeholders. The projects were all performed for medium to large-scale companies, and each project timeframe was from 1-3 years. For all these projects, the common task was to provide a problem analysis and design the system architecture design that included the integration of multiple independent systems. This was preceded by a thorough stakeholder analysis and domain analysis of the corresponding application domains. The industrial case studies from which we derived the obstacles are shown in Table 1. A short description of each of these case studies is given as follows.

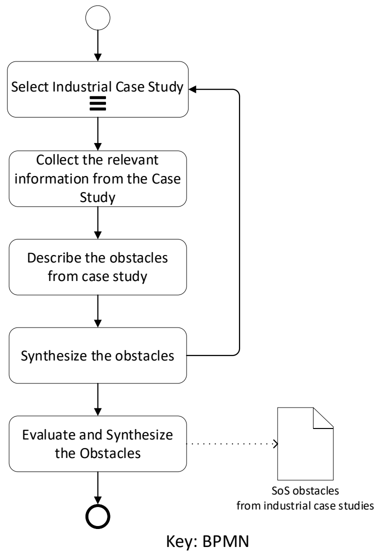

Fig. 3. Process for identifying obstacles from SoS

TABLE 1. INDUSTRIAL CASE STUDIES FROM WHICH OBSTACLES WERE IDENTIFIED

Industrial SoS Domain	**SoS Type**	**Selected Related References**
Physical Protection SoS	Technological, Directed, Organizational, Decentralized, Cyber, Complicated, Medium-Tech	[10-14]
Smart metering/Grid SoS	Technological, Acknowledged, Federated, Decentralized, Cyber, Complicated, Medium-Tech	[11]
Smart Farming SoS	Socio-Technical, Acknowledged, Federated, Decentralized, Cyber-Physical, Complicated, Medium-Tech	[15][22] [18][21]
Defense C4ISR SoS	Technological, Directed/Acknowledged, Organizational, Centralized, Cyber-Physical, Complicated, Medium-Tech	[12]
Healthcare SoS	Socio-Technical, Directed, Organizational, Decentralized, Cyber, Complicated, Medium-Tech	[19]
Enterprise SoS	Technological, Acknowledged, Organizational, Decentralized, Cyber, Complicated, Medium-Tech	[16]
Space SoS	Technological, Directed, Organizational, Centralized, Cyber-Physical, Complicated, High-Tech	[13]

1) *Physical Protection SoS*

A physical protection system (PPS) integrates people, procedures, and equipment for the protection of assets or facilities against theft, sabotage, or other malevolent intruder attacks [7][8][9]. A PPS provides deterrence and a combination of detection, delay and response measures to protect against an adversary's attempt to complete a malicious act. A PPS includes physical protection devices such as interior and exterior intrusion detection sensors, cameras, barriers, access control devices and response measures. The project dealt with developing PPS that is hierarchically structured using a system of systems approach [9][10][11].

2) Smart Metering and Smart Grid SoS

A smart meter (or advanced meter) is a device that can measure gas, heat, electric, or water consumption and submit the measured values over the network to the utility providers [11]. A smart metering system involves the use of smart meters at residential customers and the regular reading, processing and feedback of consumption data to the customer. Smart Metering is often referred to as automated meter reading (AMR), or in the case of real-time, two-way communications, as advanced metering infrastructure (AMI). Smart metering offers several capabilities including (1) real-time or near-time measurement of utility use (2) read the meter both locally and remotely (on demand) (3) interconnection to premise-based networks and devices. Within the context of the project, smart metering system was designed for the larger smart grid. A system of systems approach was needed to integrate the substituent elements that were geographically distributed and showed operational and managerial independence.

3) Smart Farming SoS

Smart farming adopts advanced technology such as cloud computing, remote sensing, artificial intelligence big data analytics and Internet of Things (IoT). One of the key elements of the smart farming is the farm management information system (FMIS) that is integrated with other systems such as farm equipment system, weather monitoring system and ERP system. Smart Farming SoS go beyond a simple record-keeping and include advanced modules supporting a comprehensive set of farming operations. The SoS supports smart and automated information gathering and merging. It helps as well as in monitoring sensor data coming from different machines, animals, plants, other farms and greenhouses and other systems such as unmanned air and land vehicles. In this way, the decision making and planning in the agricultural domain can be further supported which can lead to even more effective and efficient farming. With the help of the digital solutions, farming practices such as yield monitoring, cultivar selection, pest management, irrigation, etc. can be applied more precisely. Crop yield can be monitored and precise crop maps which show high and low production areas can be obtained readily. This SoS was considered in multiple industrial and research projects within Wageningen University.

4) C4ISR SoS

C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance) systems represent an entire system including all equipment, device, methods, procedures and personnel needed for accomplishing a mission given by Command and Control units. C4ISR terminology itself is composed of (1) Command and Control, which stands for directing military forces, (2) Communications and Computers, which are the key elements of modern warfare, and (3) Intelligence, Surveillance, Target Acquisition and Reconnaissance, which describe observing and targeting the enemy in the battle field. A typical C4ISR project works on a large-scaled distributed environment which means there are number of processes running on different machines and communicating with each other by a middleware such as CORBA, DDS (Data Distribution Service) on tightly-coupled or loosely-coupled (e.g. SOA) architectures. Furthermore, those projects should run in a real time or nearly real time manner. Therefore, most of the differences occur on application layer which manages specialized sensors, weapons, or tactical doctrines. This project was similar to existing SoS projects in the defense domain and was carried out with multiple companies.

5) Healthcare SoS

Healthcare relies on health information systems (HISs) to support various care processes and receive reimbursement for the care provided. Examples of functionalities are financial management, daily reporting, and medication management.

Because HISs consist of many interrelated software modules that should communicate, coordinate, and evolve over time. HISs assist healthcare organizations in processing data, information, and knowledge in order to contribute to high-quality, efficient patient care [19]. The HIS is considered as a building block of the complete health system, and therefore a well-functioning HIS is a vital aspect for delivering excellent care and receiving reimbursement for the care given. Within this project multiple HISs geographically distributed in The Netherlands, had to be integrated to support intellectual disability care [20].

6) Enterprise SoS

Enterprise resource planning (ERP) system is used for managing, automating, and integrating all the business functions within an organization. As such, many software firms are competing in this software component market in customer relationship management (CRM), planning & scheduling, supply chain planning, transportation and logistics markets. ERP system compounds sets of integrated software packages including business tools and applications for product planning, cost and development, manufacturing, inventory management, supply chain, marketing and sales, shipping and payment, human resources, and customer information [16]. ERP system has a shared database which supports multiple functions used by different business units. Using this shared database, an ERP system shares common data and provide access to desired data across the various departments within the organization. This indicates that ERP system affects most of the departments in an organization and as such, different stage-based approaches are preferred for the investment decisions of this type of IT systems. Traditionally, the so-called on-premise ERP systems include a shared database that supports multiple functions used by different business units. With the novel developments in distributed computing, data processing, and information systems in the last years, these traditional on-premise ERP systems have become under attack. Various studies have reported on different obstacles with the adoption of on-premise ERP systems. This part includes the experiences of industrial projects in the finance and insurance domains.

7) Space SoS

In the context of spaceflight, a satellite is an artificial object which has been intentionally placed into orbit. Such objects are sometimes called artificial satellites to distinguish them from natural satellites such as the Moon. A space system consist basically of three segments including Ground Segment, Launch Segment and Space Segment [13]. Each segment by itself is a complex system integrating hardware and software. Space systems have to deal with massive spatio-temporal Earth and Space observation data collected by space-borne and ground-based sensors. This project dealt with the system architecture design of the substituent elements of space systems (satellite systems) and the integration and alignment of these systems into a coherent SoS.

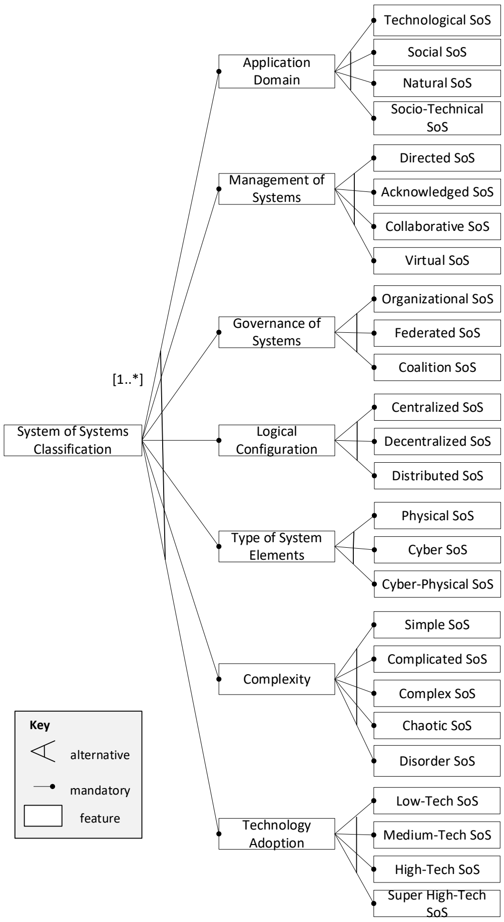

Fig. 4. Feature diagram showing the multi-dimensional classification of System of Systems (adapted from: [6])

8) *Characterization of the Industrial Case Studies*

Different types of SoSs can be distinguished. In our earlier study, we have proposed a multi-dimensional classification of SoS [6] as it is shown in Fig. 4. The figure shows a feature diagram that represents the common and variant elements of a system. The characterization of each SoS is shown in the second column of Table 1 (SoS Type). The related references are shown in the third column. It should be noted that for some projects, publications were limited due to confidentiality issues. Seven dimensions are defined: application domain, management of systems, governance of systems, logical configuration, type of system elements, complexity, and technology adoption. We refer to our earlier published paper for further details about this multi-dimensional characterization of SoS. We can observe that the above described industrial case studies present various SoSs, each with their own specific obstacles. In the following, we elaborate on the identified obstacles.

IV. Obstacles

Based on the domain analysis and the multiple industrial case studies, we could derive multiple different obstacles [5] for engineering SoSs:

A. *Decentralized Engineering*

In SoS, the substituent systems are often operational and managerial independent. As such this necessarily requires decentralization of the engineering process, including decentralized data, decentralized development, decentralized evolution and maintenance, and decentralized operation control. Yet, the identified approaches in the domain analysis process were largely focused on traditional systems engineering approaches (e.g. V model). These approaches are typically based on top-down engineering going from a requirements elicitation to the design and implementation of the system. Since an SoS is usually at a larger scale and has a broader scope than traditional systems, the traditional central engineering perspective is less feasible or no longer adequate.

B. *Continuous Evolution and Deployment*

SoSs go beyond single systems and do not only constitute multiple substituent elements but will also be in service for a longer time. It is often impractical to replace or retire SoSs altogether. Instead, SoSs will be characterized by an increasing need for evolution whereby new capabilities are deployed, and unnecessary capabilities will be given up. The analogy for this obstacle is often given using the evolution of cities which must continuously function despite ongoing maintenance and improvement activities. The evolution of SoSs is also continuous, and adaptations will be made continuously to meet the changing requirements. Further, for the adaptation process it will be necessary which changes have only a local impact and as such can be carried out concurrently, and which changes have a systemic impact over the whole SoS and as such must be coordinated. One could argue that iterative, incremental and agile methods, and with this DevOps have been proposed to cope with the evolution aspects and quick development and deployment. Unfortunately, these approaches tend to focus on single systems that do not scale yet to cope with the challenges of continuous evolution and deployment that we can observe in SoSs.

C. *Informal Modeling or Lack of SoS Modeling*

For engineering SoS, adopting the proper modeling abstractions is important to support the communication among stakeholders, guide the design decisions, and analyze the SoS with respect to the defined functional and quality criteria. An important obstacle is the lack of explicit modeling formalisms for SoS. Informal box and line drawings are often used, which are open for multiple interpretations. On the other hand, a trend toward model-based systems engineering applied to SoS in which more formal models are now being applied. Hereby SoS engineering relies on existing modeling abstractions. Both from the domain analysis and the industrial case studies, we could identify that either systems engineering (e.g., SysML) or software engineering modeling abstractions were adopted, and no explicit SoS modeling approaches were identified.

D. *Multi-Paradigm Modeling and Engineering of SoS*

SoSs are complex and dynamic systems that integrate physical, software, and network aspects. To date, no unifying theory nor systematic design methods, techniques and tools exist for such systems. Individual (mechanical, electrical, network or software) engineering disciplines offer dedicated solutions for systems in their disciplines but are limited when considering SoS. To support the engineering and integration of SoSs, a multi-paradigm modelling (MPM) approach is needed to model the SoS at the appropriate level(s) of abstraction. MPM is a research field that aims to combine different levels of abstraction and views, using modeling formalisms and semantic domains, with the goal of simulating or realizing systems. The key challenges in MPM are finding adequate modeling abstractions, multi-formalism models, and model transformations. For developing SoS multi-paradigm modeling will be necessary.

E. Socio-Technical Concerns

In SoS, people are not just users of the system but will be an active part who design, develop, use, test and maintain the system. Hence, it will be hard to understand the aspects of an SoS without full consideration of the human behavior in the system. When designing current systems, the focus tends to be on technology, and the functional and quality concerns of the technical system. A socio-technical perspective that takes into account both the technical and the human aspects is largely missing. Hereby, not only the behavior of individuals but also the collective behavior of groups of users and developers will need to be analyzed to get an insight in how the SoS is used, viewed, accepted, and maintained. A socio-technical system typically can be considered as consisting of a social subsystem and technical subsystem. The social subsystem consists of people and people in relation to each other (i.e. structure). The technical system does not include human elements and consists of technology and process. The process component defines the business process, that is, the series of steps to complete a business activity. The technology component of the technical system consists of software, hardware and the networking or telecommunications. A problem in socio-technical systems is the alignment of these components. The so-called business-IT alignment problem has been addressed largely for single systems but not yet explicitly considered at the scale of system-of-systems.

F. System-of-Systems Ecosystem

An SoS is constituted of dynamic environment of a set of interdependent sub-systems that comprise computing devices, people and organizations. In this context, we can consider many SoSs as *socio-technical ecosystems*. The concept of ecosystem is inspired from natural ecosystems in which organisms are characterized by symbiotic relationships and their survival relies heavily on the survival of the ecosystem. In the domain of software systems, a software ecosystem (SECO) is a collection of systems, which are developed and co-evolve in the same environment. Typically, a SECO consists of a common software platform and a community of internal and external actors that compose software systems to satisfy their needs. SECO-based development is different from the traditional software development in which a software product was the result of the effort of an independent software vendor typically developing a monolithic product. In software ecosystems, the development is not intra-organizational but inter-organizational and as such spread outside the traditional borders of software companies to a group of companies, private persons, or legal entities. Software ecosystems are also different from traditional outsourcing techniques since the initiating actor or platform owner does not necessarily own the software produced by contributing actors. To develop socio-technical SoSs it is important to integrate the concepts of SECO. Here it should be noted that current SECO architectures are typically focused on a single platform system with multiple developers and consumers. This does not align with the larger size and scope as is required from the SoS.

G. Design Optimization

SoS is a collection of task-oriented or dedicated systems that pool their resources and capabilities together to create a new, more complex system which offers more functionality and performance than simply the sum of the constituent systems. To provide the global level optimization while considering the local optimizations of the constituting systems in SoS, novel design optimization approaches are required. The optimization of the configuration will need to be dynamic due to the adaptable and open-ended behavior of SoS. Different quality factors need to be identified and a trade-off analysis performed to constantly tune the SoS to a feasible configuration. In particular, for the case of directed and acknowledged SoS in which tasks need are allocated over different systems it is important to search for the proper configuration to achieve the SoS level quality concerns.

H. Emergent Properties

The behavior of an SoS is not localized to any component system. An SoS has often to deal with emergent behaviors that can appear when several system components interact in complex ways. Emergent behavior is only exhibited at the global SoS level and cannot be achieved by any of the constituent systems. The SoS will need to have capabilities to respond to emergent behavior and, as such, must be able to observe their own operations, recognize acceptable and unacceptable behaviors, and take corrective action with little or no operator intervention [55]. On its turn this will require the fundamental models of SoS that are linked to the conceptual notion of emergence. In addition, corresponding tools will be needed through which operators can gain early warning of potential emergent behavior and devise strategies to deal with it.

I. Governance of SoS

Different forms of control can be applied in the acquisition and operation of the constituent systems of a SoS. The earlier defined four types of SoS also imply different kinds of governance structures. To achieve effective performance of an SoS it is important that the proper organizational structure is selected and applied. The lack of performance of an SoS can be often related to non-technical governance such as mismatches between the organization structures and the global business requirements of the SoS.

J. Heterogeneity and Interoperability

An SoS is typically a system consisting of multiple independent and heterogeneous systems. In SoS, the elements will be heterogeneous in part because they will come from a variety of sources. The independent system components will run on different hardware/software platforms, developed using different languages and designed according to different methodologies. Hence, an important concern of SoS system design, construction, and evolution is interoperability which will entail integrating heterogeneous elements and engineering perspectives. In SoS interoperability must be addressed and managed at several levels. At the technical level interoperability, will need to address the conventional syntactic interoperability and semantic interoperability. SoS requires however a broader consideration of interoperability beyond the technical level and also consider the social, organizational, and legal level to support the integration of data and processes. For this, an infrastructure will be needed that can combine development, deployment, and operational support for interoperability between organizational teams.

K. *Evaluation of SoS*

Evaluation of an SoS will be different from the evaluation of a single system. Existing verification and validation approaches will not be useful due to the scale, heterogeneity and the continuous evolution of SoSs. Moreover, socio-technical SoSs that have human beings as participants require the evaluation not of technical but also the aspects from the social, organizational and business perspectives. Approaches like agent-based modeling [67] can provide a suitable computational model for simulating the actions and interactions of the different entities in SoSs and to assess the emergent behavior that is usually a characteristic of SoSs.

L. *Ultra-Large Scale Software Engineering*

Even though systems comprise far more than software, it is software that fundamentally makes possible the required intelligence in smart systems. It is mainly for this reason that an increasing number of major businesses and industries are now dependent on software. Most of the products today either consist of software or have been developed using software. Many innovative transformations in society have also been triggered by advances in software technology. Hence, with the increasing need for smart systems, software engineering will be a crucial core competence for most organizations. It will not be possible to develop smart systems without knowledge of software engineering. Broadening the scale to the level of system-of-systems will impose further challenges on the engineering of software. Due to the changing scope and scale, current software engineering approaches for developing, deploying and operating software-intensive SoS will not suffice.

V. CONCLUSION

System of systems engineering and management is not trivial and has to cope with several obstacles. In this paper, we have reported on the obstacles that we could derive from a domain analysis to a selected set of primary studies as well as the experiences from the development of SoSs in seven different industrial projects. Most of the obstacles are shared by all types of SoS. In general, we concluded that existing design methods and abstractions do not yet provide adequate means for coping with the identified obstacles. Our future work will focus on a further exploration of potential obstacles in additional industrial case studies and broader systematic reviews.

REFERENCES

[1] M. Jamshidi, System of Systems Engineering: Innovations for the 21st Century. John Wiley and Sons, 2008.

[2] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, Feature-Oriented Domain Analysis (FODA) Feasibility Study. 1990.

[3] M. W. Maier, “Architecting Principles for Systems-of-Systems,” Syst. Eng., vol. 1, no. 4, pp. 267–284, 1998.

[4] SEBoK, “Guide to the Systems Engineering Body of Knowledge,” Guid. to Syst. Eng. Body Knowl., 2017.

[5] B. Tekinerdogan, Engineering Connected Intelligence: A Socio-Technical Perspective. Wageningen University, 2017.

[6] B. Tekinerdogan, "Multi-Dimensional Classification of System-of-Systems," 2019 14th Annual Conference System of Systems Engineering (SoSE), 2019, pp. 278-283, doi: 10.1109/SYSOSE.2019.8753841.

[7] B. Tekinerdogan, K. Özcan, S. Yagiz, I. Yakin. Systems Engineering Architecture Framework for Physical Protection Systems, International Symposium on Systems Engineering (ISSE), Vienna, Austria, October 12-14, 2020.

[8] B. Tekinerdogan, I. Yakin, S. Yagiz, K, Özcan. Product Line Architecture Design of Software-Intensive Physical Protection Systems, International Symposium on Systems Engineering (ISSE), Vienna, Austria, October 12-14, 2020.

[9] B. Tekinerdogan, K. Özcan, S. Yagiz, I. Yakin. Feature-Driven Survey of Physical Protection Systems, 11th Complex Systems Design & Management (CSD&M) conference, Paris, 2020.

[10] B. Tekinerdogan, K. Özcan, I. Yakin, S. Yagiz. Model-Based Systems Product Line Engineering of Physical Protection Systems, NCOSE 31st Annual International Symposium 2021, Jul 17, 2021 - Jul 22, 2021.

[11] B.Tekinerdogan et al. Software Product Line Engineering for Smart Metering Systems in Smart Grids, Elektromed, Technical Report, Ankara, Turkey ,2015.

[12] B. Tekinerdogan, O. Ozkose Erdogan, O. Aktug. Supporting Incremental Product Development using Multiple Product Line Architecture. Int. J. Knowl. Syst. Sci. 5 (2014): 1-16.

[13] Tekinerdogan, B., Acar, B., Cabıoğlu, Ç., Savaş, D., Vuran, N., Tekdal, Ş., & Gürsoy, Ü. (2020). Exploration of data analytics for ground segment in space systems. In B. Shishkov (Ed.), Business Modeling and Software Design - 10th International Symposium, BMSD 2020, Proceedings (pp. 352-361). Vol. 391 LNBIP). Springer. https://doi.org/10.1007/978-3-030-52306-0_25

[14] J. S. Dahmann and K. J. Baldwin, “Understanding the current state of US defense systems of systems and the implications for systems engineering,” in SysCon 2008, 2008.

[15] O. Koksal, B. Tekinerdogan, Architecture design approach for IoT-based farm management information systems, Precision Agriculture 20 (5), 926-958, 2018.

[16] S. Sancar Gozukara, B. Tekinerdogan, & C. Catal. Obstacles of On-Premise Enterprise Resource Planning Systems and Solution Directions. Journal of Computer Information Systems, 62(1), 141-152. https://doi.org/10.1080/08874417.2020.1739579, 2022.

[17] D. J. Snowden and M. E. Boone, “A leader’s framework for decision making,” Harv. Bus. Rev., 2007.

[18] J. Tummers, A. Kassahun, & B. Tekinerdogan. Reference architecture design for farm management information systems: a multi-case study approach. *Precision Agriculture*, *22*, 22-50. https://doi.org/10.1007/s11119-020-09728-0, 2021.

[19] J. Tummers B. Tekinerdogan, H. Tobi, C. Catal, B. Schalk. Obstacles and features of health information systems: A systematic literature review. Comput Biol Med. 2021.

[20] J. Tummers, H. Tobi, B. Schalk, B. Tekinerdogan, G. Leusink. State of the practice of health information systems: a survey study amongst health care professionals in intellectual disability care. BMC Health Serv Res 21, 1247 (2021).

[21] C. Verdouw, H. Sundmaeker, B. Tekinerdogan, D. Conzon, T. Montanaro. Architecture framework of IoT-based food and farm systems: A multiple case study. Computers and Electronics in Agriculture. Vol. 165:104939-. DOI: 10.1016/j.compag.2019.104939

[22] Kassahun, R. J. M. Hartog, and B. Tekinerdogan, “Realizing chain-wide transparency in meat supply chains based on global standards and a reference architecture,” Comput. Electron. Agric., 2016.