A Model-Based Approach to Document a System-of-Systems

Stephan Baumgart
Volvo Autonomous Solutions
Eskilstuna, Sweden
Email: stephan.baumgart@volvo.com

Sasikumar Punnekkat
School of Innovation, Design and Technology
Malardalen University, Sweden
Email: sasikumar.punnekkat@mdh.se

***Abstract*—The technical evolution enables the development and application of autonomous systems in various domains. In the on-road and off-road vehicle domains, autonomous vehicles are applied in different contexts. Autonomous cars are designed as single system solutions, while in other scenarios, multiple autonomous or semi-autonomous vehicles are integrated into a system-of-systems. We utilize a case from the earth-moving machinery domain, where a fleet of autonomous vehicles is used for transporting material in off-road environments. The traditional industrial development processes in the earth-moving machinery domain focus on single human-operated systems and lack clear support for autonomous system-of-systems. From our studies of industrial development of system-of-systems, we recognize the demand for guidance on how to document a system-of-systems. The goal of this work is to provide a framework using different model-based formalisms. As a structural background, we utilize the SafeSoS process, where each step specifies details about the targeted system-of-systems. Specifically, we apply model-based systems engineering to describe the structure and behavior of each SoS level. We utilize an industrial case to exemplify how model-based concepts can be applied to capture relevant information needed for designing the system-of-systems. This work provides guidelines for practitioners in developing safe system-of-systems.**

***Index Terms*—Autonomy, Model-based Development, Model-based Systems Engineering, System-of-Systems, SySML**

I. Introduction

The development of customer products requires ensuring safety in many domains, meaning that users are not at risk when using the product. The targeted products' foreseen usage scenarios need to be understood and considered during the safety analysis to ensure product safety. This is standard practice in industrial development processes today. Furthermore, products like cars, trucks, or earth-moving machinery are usually developed in generations, meaning the phased evolution of the existing products. Designing new products from scratch is rare in the industry due to high initial investment costs, among others.

Nonetheless, we can observe a paradigm shift towards utilizing autonomous features or developing entirely new autonomous solutions in many domains. We can distinguish between single system automation solutions like autonomous cars and multiple autonomous or semi-autonomous systems integrated into a system-of-systems in the vehicular domain. Integrating autonomous and human-operated machines into a system-of-systems can be observed in the earth-moving machinery domain [1]. We utilize a case from the earth-moving machinery domain for studying the development processes and industrial challenges when designing a system-of-systems.

In the earth-moving machinery domain, automation of machines enables the improvement of production workflows and increases efficiency. The constituent systems are interacting through communication channels and are reliant on the correctness of received data. Such a system can be seen as a system-of-systems. The existing development processes and standards do not support the development of system-of-systems. When considering automation and system-of-systems, the increasing complexity makes it challenging to oversee all critical scenarios and state changes.

One challenge in developing and designing complex system-of-systems is how to document all dimensions necessary in development. Model-based development is gaining importance in many domains and has clear advantages compared to a specification-based development process applied in the industry today. Model-based development formalisms can capture different product views and characteristics, such as static architecture, communication views, and product behavior. This approach seems suitable for the development of system-of-systems. The question is which model-based formalisms are suitable for various abstraction layers when designing a system-of-systems.

This work's contribution is a system-of-systems process extended with a detailed description on which documentation is suitable for designing system-of-systems.

This paper is structured as follows. We describe the background and the related work relevant to this work in section II. In the following, we provide details about the industrial case we studied and where we applied our method in section III. We provide details of the SAFESOS process in section IV and explain details on how to document a system-of-systems in section V. We discuss our results and conclude our paper in section VI.

II. Background and Related Work

This section describes the background of our work, focusing on system-of-systems and model-based systems engineering.

A. Systems vs. System-of-Systems

Since our focus is on system-of-systems, we start by clarifying the distinction between the terms systems and system-of-systems. ISO 26262 defines the term system as a "set of components or subsystems that relates at least a sensor, a controller, and an actuator with one another" [2]. A more general definition of a system provided in the standard MIL-STD-882E [3] is: "The organization of hardware, software, material, facilities, personnel, data, and services needed to perform a designated function within a stated environment with specified results." In the same standard, the term system-of-systems is defined as "a set or arrangement of interdependent systems that are related or connected to provide a given capability" [3]. The standard ISO 21841 defines that system-of-systems consists of 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" [4]. A constituent system in this context is an "independent system that forms part of a system of systems (SoS)" [4].

Specific distinguishing characteristics of system-of-systems are:

Operational independence of the elements [5], boundaries and interfaces [6]: The involved systems in a system-of-systems are independent of the integration and can operate stand-alone as well.

Managerial independence of the elements [5], [6]: The constituent systems are managed and acquired independently. A clear description of the interfaces and flow of data in the system-of-systems is required.

Evolution [5]: A system-of-systems may change in different ways over time. Features in the constituent systems can change over time. A system-of-systems may be subject to changes when constituent systems can join or disconnect. Furthermore, SoS functions may change over time as well.

Emergent behavior [5]–[7]: The behavior of the system-of-systems that arise due to the integration of systems and is not the behavior of any single constituent system is called emergent behavior.

Geographic distribution [5], operational environment [6]: The constituent systems can be geographically distributed. It is necessary to describe the operational environment and its specifics.

Maier [8] has described a commonly accepted categorization of types of SoS, using the way an SoS is organized and managed as the parameter to differentiate them. He identifies three types of SoS:

1) Directed SoS, where a master system is coordinating the slave systems in an SoS.
2) Collaborative SoS, where the constituent systems may join an SoS to fulfill the goal of the SoS
3) Virtual SoS, which has no central management or agreed purpose.

Axelsson [9] provides an extension to the existing definitions by adding the constituent systems' states.

B. Safety and System-of-Systems

In this section, we briefly discuss the literature focusing on safety in a system-of-systems. Hall-May and Kelly [10] utilize a case from the military domain and describe a system-of-systems using model-driven engineering methods and create safety argumentation using the goal structuring notation (GSN) [11]. Alexander et al. [12] propose a simulation-based hazard analysis as a possibility to handle the complexity of interactions between constituent systems. Focusing on the interfaces and potential cascading failures in a system-of-systems, Redmond described the Interface Hazard Analysis method in [13].

The compliance with existing applicable safety standards like ISO 26262 [2] in the context of system-of-systems is described by Saberi et al. [14] through a platooning case from the truck domain and propose a tailored safety lifecycle. The authors highlight that it is essential to understand potential real-life scenarios in order to be able to analyze the impact of failures and their potential cascading effects in this context. Axelsson and Kobetski [15] apply the system thinking approach STAMP [16] to analyze risks in a truck platooning case.

Compliance with applicable safety standards requires considering critical scenarios during design-time. When self-adaptive collaborating systems are integrated with a system-of-systems, and no central unit is used to coordinate the autonomous systems' activities, not all constellations and situations can be considered during design-time. Instead, safety may need to be negotiated at run-time as presented in [17].

C. Model-based Systems Engineering

"Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." [18]. Model-based Systems Engineering is growing in importance in many industrial domains and modeling languages like SysML [19] or UML [20] are widely applied.

When designing system-of-systems, various concepts are proposed in the literature depending on the focus. Acheson et al. [21] propose a model-based approach focusing on agent-based modeling. Specifically, the authors utilize SysML diagrams to describe the interactions between the agents, the states, and the agent system's architecture. East-ADL is a model-based approach applied in the automotive industry. Usually, East-ADL's focus is on single vehicles, and Chen et al. [22] have applied East-ADL in the context of system-of-systems. Specifically, the authors focus on knowledge modeling, bridging the gap between design-time modeling and operation-time knowledge. SysML is also used to capture critical requirements applications like e-health systems [23]. A model-based safety architecture framework for trains considers relevant safety standards as essential inputs and utilize the models for identifying hazards and trace risk reduction concepts [24].

III. Automated Quarry Site as a System-of-Systems

This section describes an industrial case from the earth-moving machinery domain, where autonomous vehicles operate in off-road environments, such as quarry sites, for transporting purposes. A quarry site is an open surface mine, where rocks and stones are processed. Depending on the type of material mined, the processed outputs can be gravel required for road and railway constructions or crushed limestone required for cement factories. Specifically, we use the electric site research project [1] from Volvo Construction Equipment as a case for our study, where a fleet of autonomous haulers (called HX) is transporting rocks in a quarry site. These autonomous machines operate on predefined tracks as shown in Figure 1.

The automated guided vehicles follow predefined tracks on the site. In this configuration, there are two alternative possibilities to load an HX with gravel. The first way is to utilize direct loading from the movable primary crusher (PCR), filled by an excavator (EXC). Alternatively, the HX can be loaded using a human-operated wheel loader (WL). To enable choosing which loading area is relevant, the empty HX queue at the primary decision point (MDP) until receiving a mission from the fleet control server. A loaded HX transports the material to the stationary secondary crusher (SCR) and unloads its contents. The machines are getting parked during non-operation times (PA). The electrified machines get charged at the charger (CH).

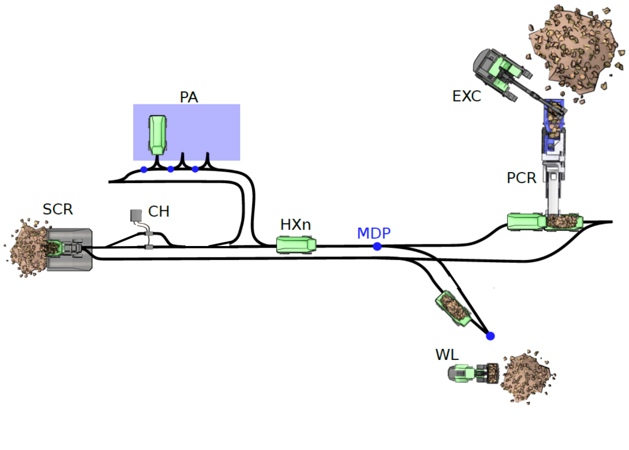

Fig. 1. Automated Quarry Site

The autonomous machines are either controlled by a central server or directly controlled using a remote control (Figure 2). There are many possibilities how an open surface mine looks like, and each mine has specific characteristics. Usually, the mine owner is defining workflows, including human-operated machines. While designing an autonomous transportation solution utilizing a fleet of autonomous machines ensuring safety during operations becomes paramount.

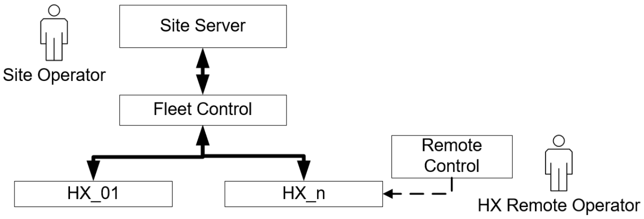

Fig. 2. Automated Quarry Site - Control Structure

Static documentation of the system-of-systems has proven to be insufficient due to the workflow and environment's dynamic characteristics, where the system-of-systems are applied. Changing weather conditions, adjusted routing, adding and removing constituent systems are just some examples when reconfiguration of the constituent systems and system-of-systems is required. Additionally, the constituent systems, the server, or infrastructure may evolve, requiring thorough analysis of the changes' impact during the operational phase to maintain the safety guarantees.

A system-of-systems is usually not limited to a single purpose. In our case, the fleet of autonomous vehicles can operate in different open surface mines. The physical characteristics differ significantly in different mines. Workflows, the type, and the number of additional vehicles integrated into the system-of-systems differ as well. Since workflows vary, the humans that work with the system or in close proximity are also specific for each mine.

We see the need for capturing the system-of-systems requirements efficiently with a low effort footprint and support for reusability from our industrial work. Traditional document-based approaches are not suitable for capturing both structural and behavior views of a system-of-systems.

IV. The SafeSoS Process

In this section, we describe an SoS-development process to support a safety analysis under consideration of the SoS as a whole together with the constituent systems' structure and behavior as presented in [25]. Our approach takes inspiration from the hierarchical SoS process described by Axelsson [9]. Our SafeSoS process contains abstraction levels for documenting an SoS as shown in Figure 3, the SoS Macro Level, the SoS Meso Level, and the SoS Micro Level. Each SafeSoS level is described below, including the significant key activities is shown in Figure 3.

As input, the system-of-systems designer needs to collect and provide information about the application scenario when designing a new system-of-systems. This method aims to store the information from each level into a database to enable reuse when designing a new system-of-systems. The stored data will require to enable reuse, potentially using product-line engineering concepts.

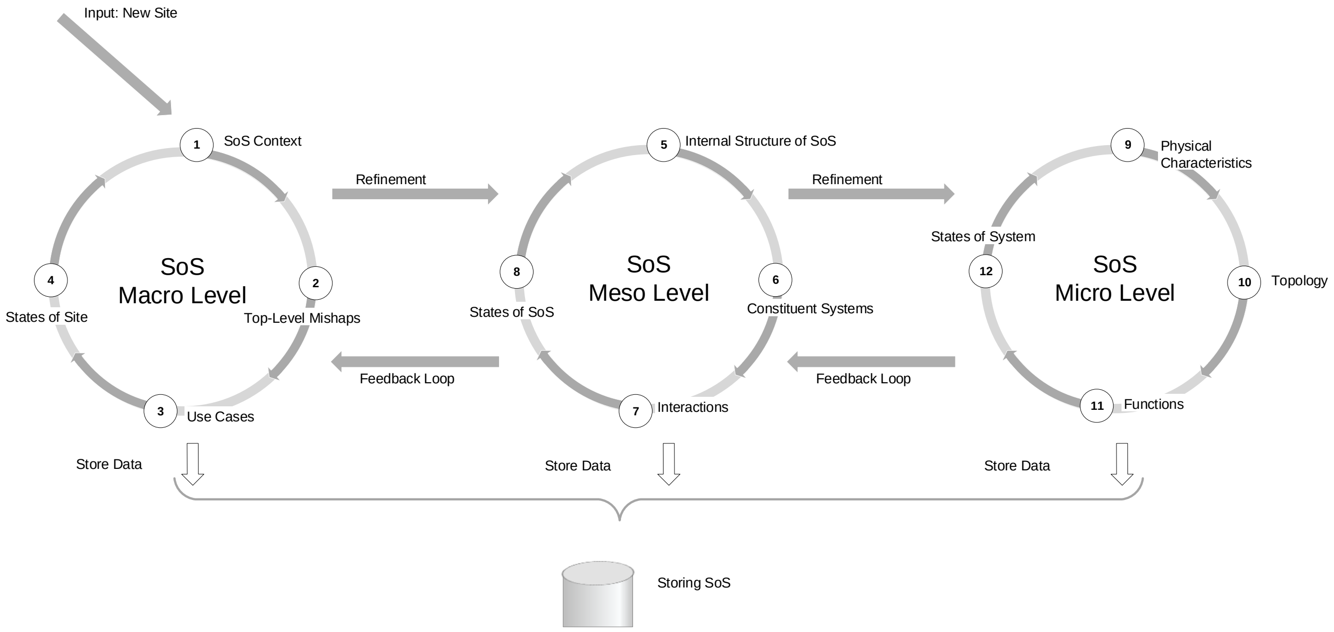

Fig. 3. SafeSoS: Safety Process to support System-of-Systems

A. *SoS Macro Level*

The SoS Macro Level's main goal is to capture the boundary of the targeted system-of-systems, environmental characteristics and derive use cases and typical scenarios.

1 SoS Context: The SoS context may contain information about the environmental conditions, the geographical characteristics, and information about the targeted workflow. Other systems that operate close to the system-of-systems need to be listed, as they can be affected or operationally limited. Humans may interact with single constituent systems or the system-of-systems as a whole. Humans also are involved in, for example, maintenance activities, where their safety is possibly affected.

2 Top-Level Mishaps: Brainstormings with stakeholders will list mishaps and losses, which will impact the design of the SoS.

3 Use Cases: Use cases and scenarios of how the system-of-systems interact with its environment, including start, pause, shutdown scenarios, are documented. Identifying potentially critical scenarios such as those with human involvement are essential outputs from this step.

4 States of the Site: The states of the SoS relate to the use cases, and critical situations that can link to inconsistent states. It is necessary to capture 'normal' operation states, safe states, inconsistent/failure states and how to recover from those.

In this initial phase, it is helpful to interview stakeholders and run brainstorming meetings with developers to understand the processes where the system-of-systems shall be applied. In such a brainstorming meeting, potential losses can be identified and rated to achieve a sorted list based on criticality. Based on the provided information, it can be analyzed which persons are at risk and which scenarios seem to be most critical. It is possible to derive hazard paths based on the identified potential losses.

B. *SoS Meso Level*

During the SoS Meso Level, the internal perspective of the SoS is in focus. This internal perspective includes the internal structure and interactions between the constituent systems.

5 Internal Structure of SoS: The internal structure of the SoS focuses on which constituent systems are participating in an SoS, possible servers, and infrastructure needed to establish the communication within the system-of-systems.

6 Constituent Systems: Details about the constituent systems and their features concerning the SoS capabilities are added.

7 Interactions: Interactions between all integrated systems need to be listed. The internal structure of the SoS(5) provides input to documenting the interactions. The interactions between humans and the constituent systems considering the humans listed in the SoS context(1) will be added.

8 States of the SoS: The states of the SoS are virtual and may not have a representation in code. Instead, the states of the SoS are the result of the constituent systems states. It can be necessary to document the states hierarchically to handle the complexity and visualize direct relations. Details about the states of the constituent systems and their dependencies shall be specified, enabling identification of safe states as well as inconsistencies.

System Designers and safety engineers can provide the required information related to the SoS Meso level. Traceability of the information provided in the SoS Macro Level will enable capturing relevant information on SoS Meso Level as we have discussed for the Interactions(7) and States(8).

C. SoS Micro Level

The SoS Micro Level contains details about a single constituent system. This level also consists of structural and behavioral views. In the SafeSoS process, we limit the scope to those single system characteristics that directly or indirectly influence the system-or-systems.

9 Physical Characteristics: The physical characteristics of a single constituent system shall be described. This may include battery properties for electrified systems, braking distance, and braking capabilities for vehicles.

10 E&E Topology: The E&E architecture or topology can be an important input to understand where the data is created and how it is shared with other systems. It is important, for example, to identify how failures in a single constituent system can cascade through the network of constituent systems and potentially lead to critical situations.

11 Functions: Apart from the topological architecture, the functions allocated to the architecture can provide additional information.

12 States of the system: A constituent system may contain several state machines on different abstraction levels. The highest level state machine is directly related to the system-of-systems.

For the Micro Level details, system developers can provide the relevant information, and safety engineers may help document all safety-related details. Changes on the SoS Micro Level may have a direct impact on the overall system-of-systems. Therefore, it is necessary to establish a structured impact analysis process.

D. Summary

The details developed in one SoS Level are refined in the following SoS level. Because information from lower SoS levels might be relevant for higher levels, a feedback loop between the levels is also envisaged. The states of the system (12) in the SoS Micro Level provide, for example, essential details to the SoS (8) states in the SoS Meso Level, where the states of the all involved constituent systems shall be captured.

V. SafeSoS - Model-based Documentation

This section provides further details on some model-based formalisms suitable for documenting an SoS in the SafeSoS process. We provide details on each level by utilizing the industrial case to explain how to document a system-of-systems using model-based documentation concepts.

A. SoS Macro Level

a) 1. SoS Context: As one of the first steps, the characteristics of the environment where the SoS shall be applied, are documented. We found it helpful to utilize SysML block diagrams and packages to document details about an open surface mine. In Figure 4, the SoS case of the open surface mine context is shown. Information that needs to be captured is:

- Systems in SoS (Vehicles): At first, the systems participating in the system-of-systems shall be listed. Additionally, systems that operate, for example, in close proximity to the system-of-systems may be added, if there is a likelihood of being affected by the system-of-systems.
  In our case, all vehicles operated within the autonomous mine boundary are collected. Apart from the autonomous vehicles (HX), other vehicles are required in the open surface mine. Typically, there are machines required for path preparation during summer or winter time. Additionally, rescue teams may be required to enter the geographical area where the system-of-systems is operating.
- Potential Exposed Humans: If humans are directly or indirectly involved in the operation of the system-of-systems or can be affected by its operation, these shall be listed. Additionally, a role description is helpful as it may limit the geographical areas of possible critical exposure. We can distinguish between educated personnel for the automated transport solution in the automation mine case, like the machine operators or the site operator and non-educated staff working with other parts of the mine. Rescue teams may need to enter the area of the automated operation. Each of the identified humans will support a risk assessment at later stages.
- Infrastructure: The constituent systems rely on communication to enable the interaction needed for fulfilling the SoS tasks. Infrastructure may contain information about the used networks and other means needed for successful operation for the system-of-systems.
  For the automation mine case, we have identified the required network infrastructure like 4G or WIFI. Additionally, a separate emergency stop network may be applied to stop the autonomous operation in critical situations. Tracks and routes are also relevant infrastructures in the automation site case.
- Environment Conditions: Environmental conditions may have an impact on the operation of the SoS. In in-house application of an SoS like in production operations, the environmental conditions may not be that relevant or obvious.
  In our case, environmental conditions are essential to document. Dusty roads during summer may affect the sensors at the constituent systems. During winter, the tracks may be icy, which may impact the traction and brake performance.
- Material to Transport: It is essential to capture the purpose of the system-of-systems. In our case, crushed rocks are

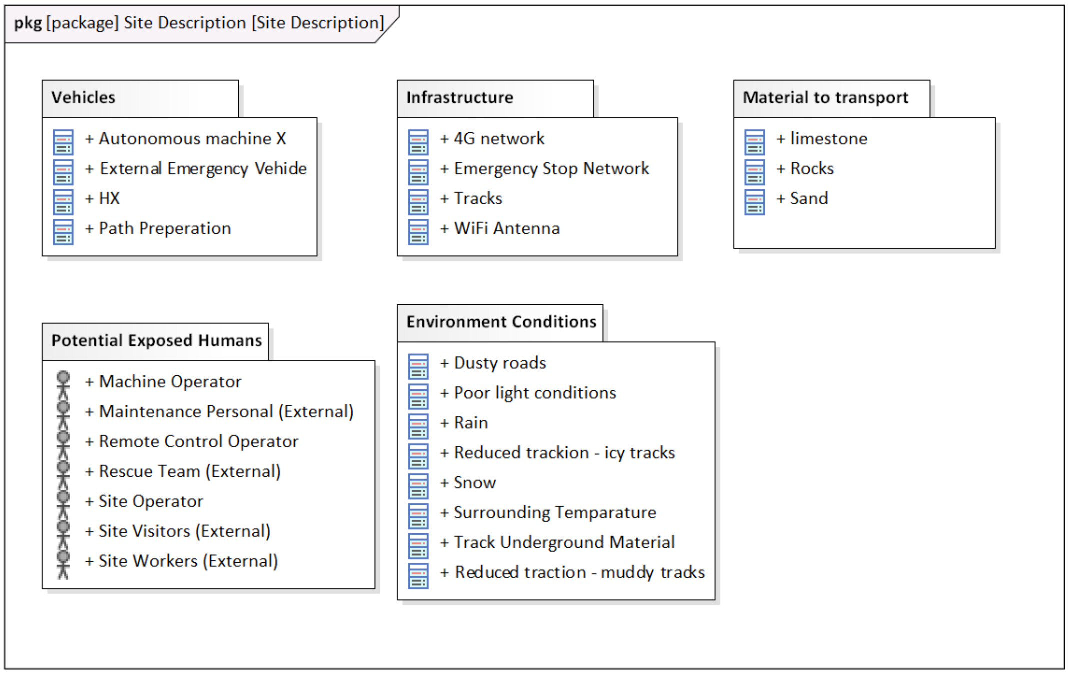

Fig. 4. SafeSoS Macro-Level: Site Description

transported by autonomous vehicles. The chemical and physical characteristics of the transported material may impact the capabilities of the system-of-systems.

b) 3. Use Cases and Scenarios: During the SoS Macro Level, use cases are captured using the SysML use case diagrams, and each use case is documented in a textual manner as described in [26]. In Figure 5, we exemplify the automated mine cases using the situation, when the autonomous machines are moved from night parking to the targeted position inside the autonomous operating zone (AOZ). Additionally it can be useful to use SysML Activity diagrams to explain procedures and processes.

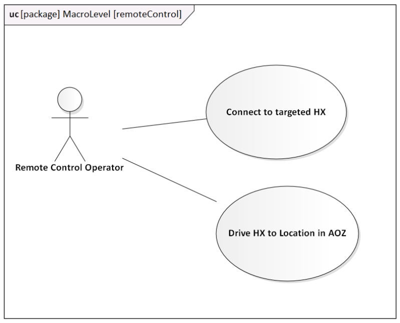
uc [package] MacroLevel [remoteControl]
Remote Control Operator
Connect to targeted HX
Drive HX to Location in AOZ

Fig. 5. SafeSoS Macro-Level: UseCase

c) 4. States of Operation: The environment in which the SoS is operated may have underlying states which may impact the operation and states of the SoS. We utilize SysML state chart diagrams for capturing those details.

In Figure 6 we provide a simplified overview of the operating phases at an open surface mine. The autonomous operation in the mine is prepared, and thereafter the autonomous fleet is activated. Once the production goal is reached, the site is stopped and shut down. In the case of a critical situation, the

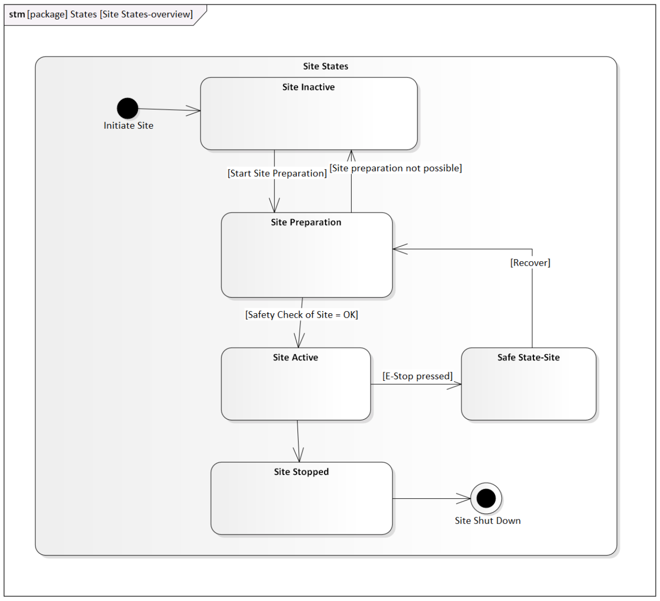

Fig. 6. SafeSoS Macro-Level: Site Process

fleet of autonomous vehicles can be set into a safe state, i.e., all autonomous vehicles stop.

B. SoS Meso Level

The purpose of the SoS Meso Level is to document details of the constituent systems in the context of their involvement in the SoS. Additionally, the interaction between the constituent systems or other actors shall be described. Apart from the SoS Meso Level's structural dimension, the interaction between the constituent systems is highlighted in the following.

For the SoS Meso Level, we exemplify two diagrams describing the behavior between the constituent systems and the involved humans. Specifically for human interaction with autonomous vehicles, we utilize SysML activity diagrams. As shown in Figure 7, we are able to add the involved humans and utilize swim lanes for depicting how the systems interact and communicate. The example shown in Figure 7 is focusing on the situation when an operator presses the emergency stop (E-Stop) button, and the fleet of HX shall stop. In this case, the E-Stop button can be pressed by the Site Operator (SO), the Excavator Operator (EXOP), or the Wheel Loader Operator (WLOP). The Safe state, as shown in the SoS Macro Level, is reached for the system-of-systems. Additionally, the HX will stop and switch the state and engage the parking brake. This indicates the direct connection between all SafeSoS levels.

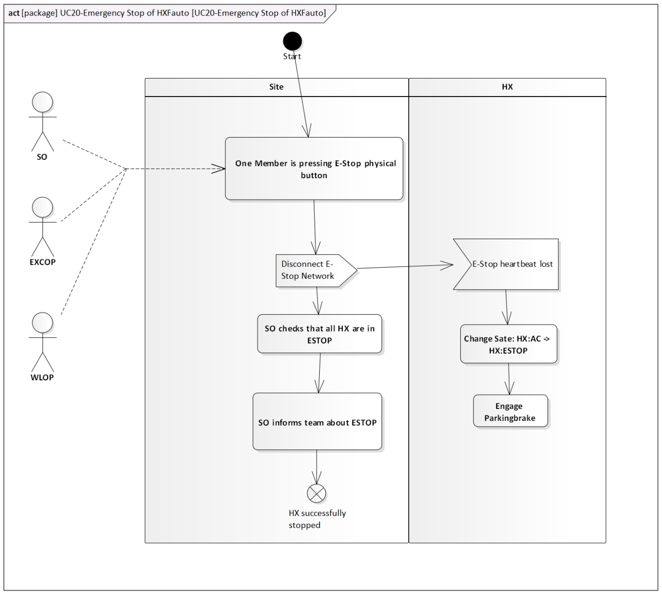
act [package] UC20-Emergency Stop of HXFauto [UC20-Emergency Stop of HXFauto]
Start
Site
HX
SO
EXCOP
WLOP
One Member is pressing E-Stop physical button
Disconnect E-Stop Network
E-Stop heartbeat lost
SO checks that all HX are in ESTOP
Change Sate: HX:AC -> HX:ESTOP
SO informs team about ESTOP
Engage Parkingbrake
HX successfully stopped

Fig. 7. SafeSoS Meso-Level: E-Stop

We utilize SysML sequence charts for the communication between the constituent systems as shown in Figure 8. Each constituent system is put into a separate swim lane.

In this simplified example, the fleet of HX is running autonomously and controlled by the server, and the remote control operator requests to receive control for a specific HX. The single HX must negotiate which commands it should listen to, either from the server or from the remote control. If the fleet control server approves the request, the HX is removed from autonomous operation, and the Remote control operator is receiving control

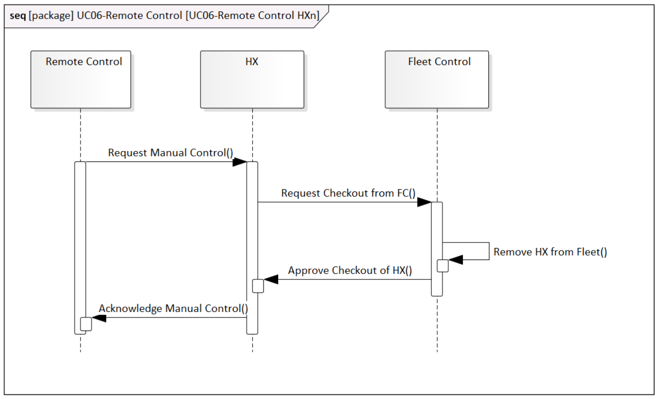
seq [package] UC06-Remote Control [UC06-Remote Control HXn]
Remote Control
HX
Fleet Control
Request Manual Control()
Request Checkout from FC()
Remove HX from Fleet()
Approve Checkout of HX()
Acknowledge Manual Control()

Fig. 8. SafeSoS Meso-Level: Request Remote Control from Server

C. SoS Micro Level

The SoS Micro Level contains details about each specific constituent system. We exemplify how the details can be captured in Figure 9, where the states of an autonomous HX are documented using a SySML state chart diagrams. The figure shows a simplified model of the actual states. The machine can be in standby mode, which directly relates to the startup state of the SoS as shown in the SoS Macro Level above. Once the machine is started, it is set into standby mode. Changing state requires human interaction with the remote control as described in the SoS Meso Level above. When the HX is driven into the targeted position with the remote control, the Site Server can control the HX if the Site Operator approves the autonomous operation. The Site Operator can

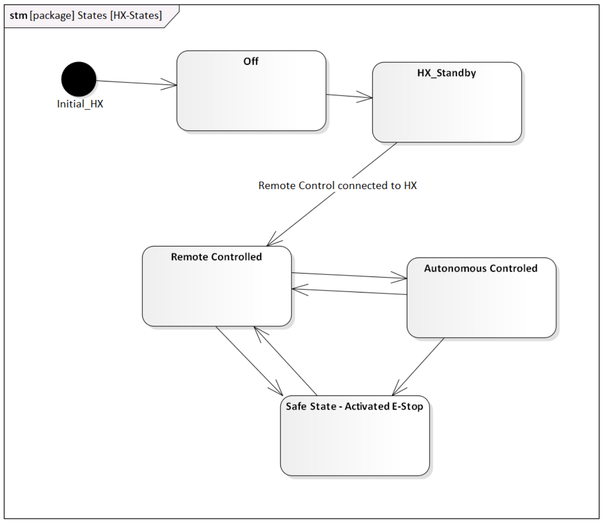
stm [package] States [HX-States]
Initial_HX
Off
HX_Standby
Remote Control connected to HX
Remote Controlled
Autonomous Controled
Safe State - Activated E-Stop

Fig. 9. SafeSoS Micro-Level:States of HX

take over an autonomous HX as shown in Figure 8 in the SOS Meso Level, which would lead to a state transition *Autonomous Controlled* to *Remote Controlled*. In case of an Emergency stop pressed, the transition to the SoS safe state will lead to that each HX will be transferred to its safe state. There are more state machines on a single machine that all directly connected. Additionally, we have shown how the states are related to other SafeSoS Levels above. Identifying all dependencies is challenging, and it will be necessary to utilize, for example, formal verification methods to show freedom from deadlocks or undiscovered inconsistencies as we have presented in [27].

VI. Discussion and Conclusion

Automation of systems and processes grows in importance in many domains. We focus our work on system-of-systems, where several systems are combined to realize a process or workflow. We utilize a case from the earth-moving machinery domain, where a fleet of autonomous vehicles is used for transporting material in off-road environments. This autonomous fleet is not operated in a stand-alone mode. Instead, other human-operated vehicles or staff may be involved in the workflows in an open surface mine. The traditional industrial development processes in the earth-moving machinery domain focus on single human-operated systems and lack clear support for autonomous system-of-systems. In this work, we have described the SafeSoS process, a hierarchical approach to documenting a system-of-systems' characteristics and properties. Specifically, we utilized model-based systems engineering techniques such as SysML models to capture the structure and behavior on each SafeSoS level. The models on the different SafeSoS Levels are connected and may impact each other. The states of the HX are, for example, not independent from the SoS, where it is integrated and vice versa.

Further research is required to enable clear traceability between the models and the levels. A software tool can support managing the complexity and visualize and report inconsistencies between the different SafeSoS Levels.

Acknowledgment

The authors acknowledge the funding support received for this research from the KKS-funded ITS-EASY Post Graduate School for Embedded Software and Systems, the ESCAPE Project funded by Vinnova and the SUCCESS Project supported by the Assuring Autonomy International Programme (AAIP), a partnership between Lloyd's Register Foundation and the University of York.

References

[1] Volvo Construction Equipment, "Electric Site Project," 2. [Online]. Available: https://www.volvoce.com/global/en/news-and-events/news-and-press-releases/2018/carbon-emissions-reduced-by-98-at-volvo-construction-equipment-and-skanskas-electric-site/

[2] International Organization for Standardization, "ISO 26262:2018 - Road vehicles – Functional safety," 2018.

[3] United States Department of Defense, "MIL-STD-882E," Washington, DC, USA, 2012.

[4] International Organization for Standardization, "ISO/IEC/IEEE 21841 Systems and software engineering — Taxonomy of systems of systems," 2019.

[5] M. W. Maier, "Architecting Principles for Systems-of-Systems," *INCOSE International Symposium*, vol. 6, no. 1, pp. 565–573, 1996.

[6] J. S. Dahmann and K. J. Baldwin, "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering," in *2008 2nd Annual IEEE Systems Conference*, 2008.

[7] J. Boardman and B. Sauser, "System of Systems - The meaning of of," *Proceedings 2006 IEEE/SMC International Conference on System of Systems Engineering*, vol. 2006, no. April, pp. 118–123, 2006.

[8] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[9] J. Axelsson, "A Refined Terminology on System-of-Systems Substructure and Constituent System States," *2019 14th Annual Conference System of Systems Engineering (SoSE)*, pp. 31–36, 2019.

[10] M. Hall-May and T. Kelly, "Using Agent-based Modelling Approaches to Support the Development of Safety Policy for Systems of Systems," *Proceedings of the 25th International Conference on Computer Safety, Reliability and Security (SAFECOMP '06)*, pp. 330–343, 2006.

[11] T. P. Kelly, "Arguing Safety – A Systematic Approach to Managing Safety Cases," Ph.D. dissertation, University of York, 1998.

[12] R. Alexander, D. Kazakov, and T. Kelly, "System of Systems Hazard Analysis Using Simulation and Machine Learning," pp. 1–14, 2006.

[13] P. J. Redmond, J. B. Michael, and P. V. Shebalin, "Interface hazard analysis for system of systems," in *2008 IEEE International Conference on System of Systems Engineering*. IEEE, 6 2008, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/4724202/

[14] A. K. Saberi, E. Barbier, F. Benders, and M. Van Den Brand, "On functional safety methods: A system of systems approach," in *12th Annual IEEE International Systems Conference, SysCon 2018 - Proceedings*, 2018, pp. 1–6.

[15] J. Axelsson and A. Kobetski, "Towards a risk analysis method for systems-of-systems based on systems thinking," in *2018 Annual IEEE International Systems Conference (SysCon)*. IEEE, 4 2018, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8369501/

[16] N. G. Leveson and J. P. Thomas, *STPA Handbook*, 2018.

[17] D. Schneider and M. Trapp, "Runtime Safety Models in Open Systems of Systems," *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 455–460, 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5380438/

[18] International Council on Systems Engineering (INCOSE), "SYSTEMS ENGINEERING VISION 2020," Tech. Rep., 2007.

[19] Object Management Group, "SysML-Systems Modeling Language." [Online]. Available: https://sysml.org/

[20] ——, "UML - Unified Modeling Language." [Online]. Available: https://www.uml.org/

[21] P. Acheson, C. Dagli, and N. Kilicay-Ergin, "Model based systems engineering for system of systems using agent-based modeling," *Procedia Computer Science*, vol. 16, pp. 11–19, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2013.01.002

[22] D. Chen, K. Meinke, K. Ostberg, F. Asplund, and C. Baumann, "A knowledge-in-the-loop approach to integrated safety & security for cooperative system-of-systems," in *2015 IEEE 7th International Conference on Intelligent Computing and Information Systems, ICICIS 2015*. Institute of Electrical and Electronics Engineers Inc., 2 2016, pp. 13–20.

[23] C. Kotronis, M. Nikolaidou, G. Dimitrakopoulos, D. Anagnostopoulos, A. Amira, and F. Bensaali, "A model-based approach for managing criticality requirements in e-health IoT systems," in *2018 13th System of Systems Engineering Conference, SoSE 2018*. Institute of Electrical and Electronics Engineers Inc., 8 2018, pp. 60–67.

[24] K. Schuitemaker, J. G. Braakhuis, and M. Rajabalinejad, "A model based safety architecture framework for Dutch high speed train lines," *10th International Conference on System of Systems Engineering (SoSE)*, pp. 24–29, 2015.

[25] S. Baumgart, J. Fröberg, and S. Punnekkat, "A Process to Support Safety Analysis for a System-of-Systems," in *The 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020.

[26] G. Schneider and P. J. Winters, *Applying Use Cases: A Practical Guide, Second Edition*, 2001.

[27] S. Baumgart, J. Fröberg, and S. Punnekkat, "A State-based Extension to STPA for Safety-Critical System-of-Systems," in *4th International Conference on System Reliability and Safety*, 11 2019. [Online]. Available: http://www.es.mdh.se/publications/5674-