A State-based Extension to STPA for Safety-Critical System-of-Systems

Stephan Baumgart*, Joakim Fröberg†‡, Sasikumar Punnekkat‡
* System Architecture Department, Volvo Construction Equipment, Eskilstuna, Sweden
Email: stephan.baumgart@volvo.com
† Research Institutes of Sweden, RISE ICT/SICS Västerås, Sweden
Email: joakim.froberg@ri.se
‡School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
Email: sasikumar.punnekkat@mdh.se

***Abstract*—Automation of earth moving machinery enables improving existing production workflows in various applications like surface mines, material handling operations or material transporting. Such connected and collaborating autonomous machines can be seen as a system-of-systems. It is not yet clear how to consider safety during the development of such system-of-systems (SoS). One potentially useful approach to analyze the safety for complex systems is the System Theoretic Process Analysis (STPA). However, STPA is essentially suitable to static monolithic systems and lacks the ability to deal with emergent and dysfunctional behaviors in the case of SoS. These behaviors if not identified could potentially lead to hazards and it is important to provide mechanisms for SoS developers/integrators to capture such critical situations. In this paper, we present an approach for enriching STPA to provide the ability to check whether the distributed constituent systems of a SoS have a consistent perspective of the global state which is necessary to ensure safety. In other words, these checks must be capable at least to identify and highlight inconsistencies that can lead to critical situations. We describe the above approach by taking a specific case of state change related issues that could potentially be missed by STPA by looking at an industrial case. By applying Petri nets, we show that possible critical situations related to state changes are not identified by STPA. In this context we also propose a model-based extension to STPA and show how our new process could function in tandem with STPA.**

***Index Terms*—Hazard Analysis and Risk Assessment, System-of-Systems, Autonomous Machines, STPA, Safety, Petri Net**

I. Introduction

Developing safety critical products requires to understand how the targeted customers use the products. This will help to identify those situations where human errors or failures in the involved systems may lead to critical accidents. Apart from focusing on features to avoid accidents or to reduce the impact of accidents, functional safety focuses on designing the electrical and electronic system (E/E) in such a way that faults in the E/E system will not lead to accidents and the system is put into a safe state. Considering functional safety during the development requires rigor in following development processes defined in the functional safety standards. These standards help developers to avoid critical systematic failures in software and random failures in hardware. Domain specific functional safety standards like ISO 26262 [1] for the automotive domain, ISO 13849 [2] or ISO 19014 [3] for the earth moving machinery domain or the generic functional safety standard IEC 61508 [4] provide guidance for ensuring functional safety during development of safety critical products. As an initial phase, potential hazards related to a product need to be identified and analyzed. Hazard analysis methods applied in development processes in industry are for example Preliminary Hazard Analysis (PHA) [5], Hazard and Risk Assessment (HARA) [1], Fault Tree Analysis (FTA) [6] and Failure Mode and Effect Analysis (FMEA) [7]. PHA and HARA are applied during early phases in the development process to list and evaluate possible hazards related to the product to be developed. FTA and FMEA are applied during later stages in the development process as they require detailed knowledge about the targeted architecture and the used components. The processes described in the functional safety standards as well as the established hazard analysis methods focus on single, human operated machines in the example of earth moving machinery. Currently, there is a paradigm shift in many domains towards adding automation to aid drivers, increase productivity and reduce risks by eliminating human errors. In the earth moving machinery domain, automation of machines enables the improvement of production workflows and the increase the efficiency as it has been shown in the Electric Site Research Project [8]. In this project a fleet of eight autonomous haulers (called HX) are utilized to transport material in an open surface quarry mine. A central server coordinates the fleet of HX and provides missions to each single HX depending on relevant site and individual scenarios. These machines collaborate to achieve common tasks, e.g. transporting material in the quarry site. Additionally, other human-operated machine can be used to interact with the autonomous machines. Such a system can be seen as a system-of-systems. A system-of-systems is defined in [9] as a "system that has operational and managerial independence of its elements." This means, that the involved systems of a SoS must be able to be operated independent from the SoS to provide a useful purpose. With managerial independence the author emphasize that the involved systems can be "separately acquired and integrated". Periorellis et al. [10] describe that "the purpose of a SoS is to provide a set of enhanced or improved "emergent" services, based on some or all of the

services provided by the participating component systems. The provision of these emergent services requires co-operation between the systems." The term system-of-systems (SoS) implies that these individual systems can be grouped and connected to provide services not achievable by one single system alone. System-of-systems rely on communicating between the independent and geographical distributed systems as failing of communication, providing erroneous data or misinterpreting correct data may lead to accidents [11].

One potentially useful approach to analyze the safety for complex systems is the System Theoretic Process Analysis (STPA) [12], which we apply to an industrial case for system-of-systems from the earth moving machinery domain in the scope of this paper. In order to identify all critical situations for a system-of-systems, such a method must be able to deal with emergent and dysfunctional behaviors of a SoS. The objective of this paper is to present an approach for enriching STPA to provide the ability to check whether the distributed constituent systems of a SoS have a consistent perspective of the global state which is necessary to ensure safety.

This paper is structured as follows. We describe the background of our paper in section II. In section III we present a case of automated machines from the earth moving machinery domain. The related work is described in section IV. We describe STPA in section V and apply it to our industrial case. We present an enhancement for STPA to additionally identify inconsistencies in system-of-systems by using Petri nets in section VI. By applying our proposed enhancement to the industrial case, we show how additional critical situations can be found. We analyze and discuss our results in section VII and conclude our paper in section VIII.

II. Background

In this section we provide background information to our work.

A. Hazard Analysis

Hazard analysis methods can be distinguished into two major groups. The first group contains methods that aim to identify and evaluate hazards during early development phases. Typical examples are PHA [13], HARA [1], the Machine Control System Safety Analysis (MCSSA) [3] or the Hazard and Operability Studies (HAZOP) [14]. Each of these methods requires in the first stage to identify the main function of the product that shall be developed. As a second stage the foreseen operation modes shall be identified as for example Idling, Working or Maintenance for the earth moving machinery domain. In brainstorming meetings with experts each operational mode will be analyzed how a failing of a function may lead to accidents. Guide words as proposed by HAZOP can provide further structure to such an analysis. Each identified hazard will be rated by estimating the severity of the accident, the probability this failure could happen and if the humans involved have the possibility to avoid the accident to happen by the controls available. The resulting estimates are used to calculate a rating of a hazard, i.e. SIL [4], ASIL [1] or PL [2], which is necessary for tailoring the development processes required by the functional safety standards. The second group of hazard analysis methods is applied during development to trace the identified top-level hazards and analyze the used architecture and components. Typical examples in this group are the top-down analysis method FTA [6] or the bottom-up analysis method FMEA [15]. FTA is using a tree structure, where the root node is a top-level failure that shall be avoided and the leave nodes are representing components in the architecture of the system to be developed [16]. FMEA is a safety analysis method, which is using a table to list all safety related components of a system. Typically, FTA provides a list of components to be analyzed. During a FMEA failing of each component is analyzed and if this can lead to system failures. The identified critical component failures are rated in the first stage and potential risk mitigation are identified. The FMEA is repeated to analyze if the applied counter measures will lead to the required risk reduction. When designing complex system-of-systems as in our case, we are interested in hazard analysis methods that are able to deal with emergent and dysfunctional behaviors in a SoS.

B. Petri nets

Various concept of modeling system specifications and system behavior are available. The goal of our work is to be able to model the states of the involved systems of a SoS and to simulate the interactions to find possible critical scenarios. Petri nets for example provide these required properties. Petri nets (PN) represent a "formal model of information flow" [17]. The graphical representation consists of places (P) depicted as circles and transitions (T) depicted as rectangles. Places and transitions "are connected by directed arcs from places to transitions and from transitions to places." [17] A transition has inputs, when arcs point from places to a transition and outputs where the arcs point from a transition to a place. The behavior of a PN is modeled by using markers which are depicted as dots on the places (P). A transition is "consuming" a predefined number of markers from an input and is generating a predefined number of markers in the output places of this transition. If it is required to simulate timing properties, Timed Petri nets [18] or Stochastic Petri nets [19] can be used. The time a transition needs for transforming the markers from the input places and generating the output markers can be defined. Timing is in our case important since a delayed communication for example may lead to critical situations. In the context of safety critical systems, El Koursi et al. [20] highlight that Petri nets can be used for modeling system specifications to check completeness and consistency and to use simulation to check correctness of safety criteria. We are specifically interested in simulating the behavior of complex system-of-systems to find possible design flaws.

III. Industrial Case - Electric Site

We utilize the electric site research project [8] as a case for our work. In this project a fleet of automated guided vehicles (AGVs) [21] called HX are used to transport material at a

quarry site, which is a surface mine for gravel production in our case. The pre-crushed material is transported from a movable primary crusher to a stationary secondary crusher. Along with the fleet of autonomous HX, a human-operated wheel loader and a human-operated excavator are used for loading material onto the HX. In our earlier work we have described and analyzed this complex SoS [22], [23].

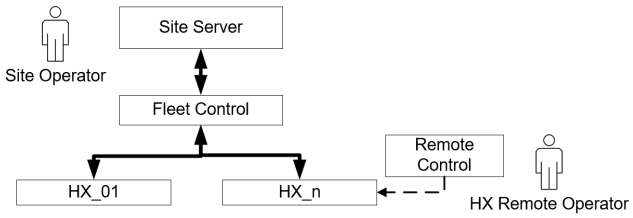

Fig. 1. Use Case: Remote Control of HX

The fleet of active HX is controlled by the Fleet Control System, containing features like traffic management or setting missions for each active HX. Each HX is therefore highly dependent on the wireless network and correct commands. In order to be able to activate a HX in the morning, remove a HX for repair purposes or adding a HX to a running production, it is possible at any given instance to control a single HX using a remote control by a HX Remote Operator. The Site Operator is monitoring the quarry site from a control room, where the Site Server is located. In Figure 1 the involved systems and human operators are presented. When designing such a system an in-depth analysis of this scenario is necessary to identify potential hazards leading to critical accidents.

IV. Related Work

We are interested in hazard analysis methods specifically considering system-of-systems and providing support for designing such a system. New approaches have been proposed to analyze hazards for system-of-systems like the System-of-Systems Hazard Analysis (SoSHA) [5], the Interface Hazard Analysis Method [24] or methods utilizing simulations to identify hazards like the Simulation based Hazard Analysis (SimHazan) [25]. These hazard analysis methods assume an integration of existing and already safety certified systems into a system-of-systems. When integrating existing systems into a compound of systems, it is necessary to ensure a safe integration. Furthermore, in many cases human operated machines are integrated into a system-of-systems. The Interface Hazard Analysis Method is focusing for example on the communication channels between the involved systems. In our case, we are designing a system-of-systems including a fleet of autonomous machines. Emergent hazards as described in the taxonomy provided by Redmond [11] may be missed when only considering safety for each single machine.

Instead, we searched for hazard analysis methods supporting the design process of complex safety critical systems. In this process it is important that analysis results are available during early stages in development process to support decision making. We focus in our work on the System Theoretic Process Analysis (STPA), which is a recent approach to analyze safety-critical systems and has grown attention [12], [23], [26]–[30]. We have attempted to use STPA for the Electric Site use case from the earth moving machinery domain [23]. During this exercise, we found several open challenges not clearly solvable by a straight-forward application of STPA and our current research focuses on making the safety analysis efficient by solving those challenges. STPA is aiming to provide inputs during early stages in the development but is not directly considering a quantification of hazards as required by the functional safety standards and as other hazard analysis methods do. Zhang et al. [31] propose an extension to STPA, called STPA-RAM, which adds quantification of identified losses to STPA in order to reduce the number of unsafe control actions and to provide guidance for decision making in industrial projects. The authors utilize Stochastic Petri nets to simulate events and use reliability data from an existing database to calculate the frequency of losses for different cases. Zhu et al. [32] apply Petri nets to formalize the control structure diagram in STPA to support the identification of unsafe control actions and their causal factors. The authors propose a new method called Control Logic Petri Net (CLPN), which is including a Petri net notation and an analysis part to find unsafe control actions. In comparison we are not aiming to replace the control structure diagram of STPA, instead we add a dimension that enables the identification and analysis of inconsistencies in the involved systems of a SoS.

V. System-Theoretic Process Analysis - STPA

To illustrate the application of STPA, we analyze the remote control case and follow the STPA process as described in literature [12].

A. STPA Overview

At first we provide a short description of STPA. STPA consists of four steps as shown in Figure 2, which we describe in the following section.

STPA - Step 1: During the first step of STPA, the scope of the STPA is set and potential losses and hazards shall be identified. System-level hazards may be derived in brainstorming meetings with experts or by applying hazard identification methods like HAZOP or What-if Analysis. The list of possible system hazards may be extended during later stages when more product knowledge is available.

STPA - Step 2: In Step 2, the control structure of the system is derived. The control structure diagram is a graphical representation of the control actions to aid a structured analysis. The control structure diagram contains the main control elements and control actions between the controllers and the controlled systems.

STPA - Step 3: The control structure diagram is used to apply a structured analysis of each control action and if a failure of the control action would lead to the already listed

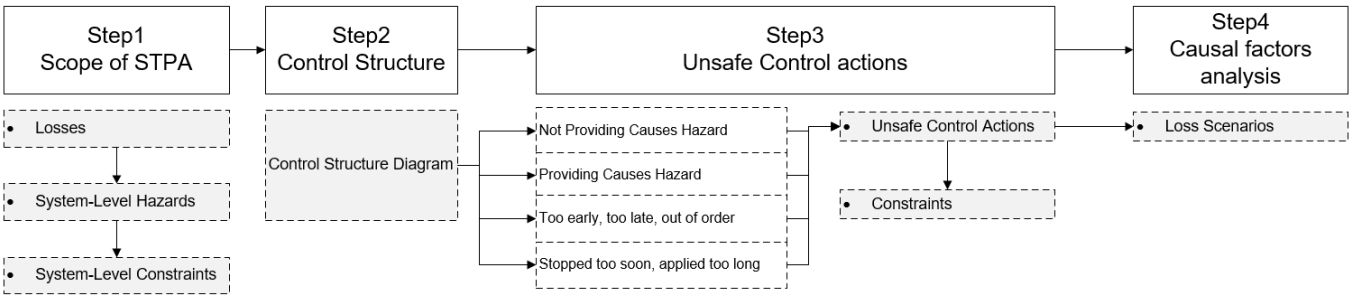

Fig. 2. General STPA Process as described in [12]

system-level hazards. STPA uses four guide words for finding such unsafe control actions:

- Not providing causes hazard
- Providing causes hazard
- Too early, too late, out of order
- Stopped too soon, applied too long

This means that the following requirements are tested:

- A correct control action is provided.
- A control action is provided at the correct time.
- A control action is provided with correct duration.

STPA - Step 4: In the last step of STPA, possible loss scenarios are identified for each unsafe control action. Reasoning why an unsafe control action would occur and how this could lead to a hazard shall be provided.

STPA - Conclusion STPA is useful for identifying and analyzing control actions and their causal factors when unsafe control actions are identified. The process of STPA is foreseen to be iterative, i.e. it is possible that further system-level or subsystem-level hazards will be identified during later stages. It is furthermore proposed to add complexity to the control structure diagram during later stages of the development process. This will lead to additional efforts for identifying unsafe control actions in Step 3.

The question is, if STPA is able to deal with emergent and dysfunctional behaviors in the case of system-of-systems. These behaviors if not identified could potentially lead to hazards and it is important to provide mechanisms for SoS developers/integrators to capture such critical situations.

B. STPA - Application Remote Control Case

In the following we apply STPA to the industrial case described above in section III.

STPA Step 1 - Remote Control Case: For our limited case we have identified two major losses that shall be avoided:

- Loss1: Humans injured or killed
  Situations, where humans are at risk to be injured or killed by the autonomous machines shall be avoided.
- Loss2: Damage of Equipment
  If machines are damaged because of accidents, this may result in a stop of production at the site, which shall be avoided.

Typical SoS hazards in our case can be:

- Hazard 1 (H-1): HX does not maintain safe distance to humans on Site.
- Hazard 2 (H-2): HX enters dangerous area/region
- Hazard 3 (H-3): Squeezing Hazard (e.g. people close to HX)
- Hazard 4 (H-4): Insufficient ability of machinery to be slowed down, stopped and immobilized

STPA Step 2 - Remote Control Case: We simplified the control structure diagram for the purpose of this paper as shown in Figure 3.

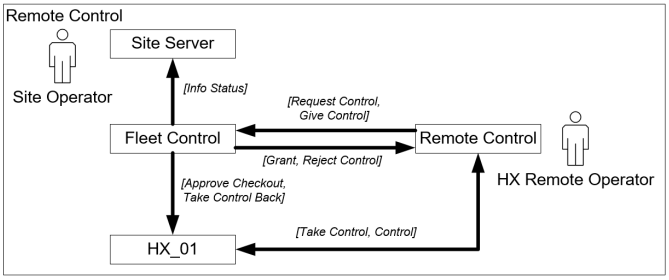

Fig. 3. Control Structure Diagram: Remote Control HX_01

The HX Remote Operator sends a request to the Fleet Control server with the purpose to take over the control of a specific HX (HX_01). Fleet Control can decide either to accept (Grant Control) or to reject (Reject Control) the request. At the same time the Fleet Control is sharing information about the active HX with the site server shown by the message Info Status. If the remote control request is accepted, Fleet Control is sending a task (Approve Checkout) to HX_01 to enable the HX to be controlled by the Remote Control. Once this is done, the HX Remote Operator can take control over the HX. The HX Remote Operator can also give back control of HX_01 to Fleet Control. Fleet Control will send a request (Take Control Back) to HX_01 that it will listen to controls send from Fleet Control.

STPA Step 3 - Remote Control Case: Each message in the control structure diagram (Fig. 3) is analyzed using the guide words.

We exemplify identifying unsafe control actions by analyzing the messages “Request Control” and “Approve Checkout” in table I. Applying the first guide word *Not providing causes*

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long
Request Control	Request Control is not provided to Fleet Control [Not Hazardous]	UCA_01: Request Control is sent unintendedly during normal operation. [H-2, H-3]	Request from HX Remote Operator is provided too late. [Not Hazardous]	
Approve Checkout	Approve Checkout is not provided to HX. [Not Hazardous]	UCA_02: Approve Checkout is provided unintended to HX during normal operation. [H-1, H-2, H-3, H-4]		

TABLE I
UNSAFE CONTROL ACTIONS: REMOTE CONTROL CASE

hazard for "Request Control" helps finding the critical situations if the message is either not provided or lost, but this will not directly lead to a hazard. We identify the first unsafe control action (UCA_01) in the situation when the message "Request Control" is provided unintended. This may lead to a situation that a HX is checked out from Fleet Control without awareness of the HX Remote Operator. Humans are at risk, if the machine is moving into dangerous areas, where humans are working (H-2) or if humans are already close by, this may lead to squeezing hazards (H-3). If the signal is delayed (Too early, too late, out of order), this may lead in the worst case to frustration of the operator, but not to hazardous situations. The message "Approve Checkout" is send from the Fleet Control to the HX to indicate, that the HX shall change mode to be controlled by a remote control. We identify, that providing "Approve Checkout" unintended, will lead to a situation where the HX is forced to switch over to be remote controlled. This can lead to critical situations where the HX is moving without a control instance connected to the machine. Altogether, we have identified 15 UCAs for this simplified case during the first brainstorming (Fig. 3).

STPA Step 4 - Remote Control Case: In our case, "Approve Checkout" might be provided unintended because of a fault in the Fleet Control software or due to a transmission error.

C. Conclusion STPA Case Study

Where is STPA suitable? STPA is a useful approach to analyze the safety of complex systems. While hazard analysis methods like PHA, FTA and FMEA focus on failures of system functions and their impact, is STPA analyzing possible failures of control actions between the involved systems and sub-systems. This analysis leads to a broader list of possible critical scenarios that require further analysis to list all causal factors. STPA is analyzing the control actions and therefore mostly communication related hazards will be identified.

Which critical situations are not captured in STPA? STPA analyzes one single control action a time, which makes it impossible to find critical scenarios which involve for example a combination of control actions, cascading failures or state changes. STPA is essentially suitable to static monolithic systems and lacks the ability to deal with emergent and dysfunctional behaviors in the case of SoS. These behaviors if not identified could potentially lead to hazards and it is important to provide mechanisms for SoS developers/integrators to capture such critical situations. It is among others important to check, if the involved systems in a SoS have a consistent perspective of the global state. The states of the involved systems are not considered in the control structure diagram of STPA. Design flaws and casual factors might be missed, if the interaction of state machines is not considered during analysis of the SoS.

VI. STPA ENHANCEMENT

In this section, we present an approach for enriching STPA to provide the ability to check whether the distributed constituent systems of a SoS have a consistent perspective of the global state. In Figure 4 we present our proposed enhancement of STPA. We exemplify three challenges regarding SoS, which require additional analysis efforts:

- **Challenge 1:** Inconsistent states in SoS
  We need to be able to even consider the states of the involved systems in a SoS.
- **Challenge 2:** Communication deadlocks in SoS
  When analyzing single messages and control actions at a time, it might not be possible to identify if seemingly correct communication will lead to a deadlock.
- **Challenge 3:** Reachability of Safe States
  When safe states are already considered, it needs to be checked and analyzed, if specific states can be reached or not. Because states are not considered in the standard STPA analysis, this need to be added.

As shown in Figure 4 we foresee additional formalisms and tools to support an analysis for SoS. Such a method is for example Petri nets, which we apply in the following to identify inconsistencies in a SoS.

A. Petri nets - Identifying Critical State Changes

We apply Petri nets to model the states of the involved systems and simulate state changes and analyze if this may

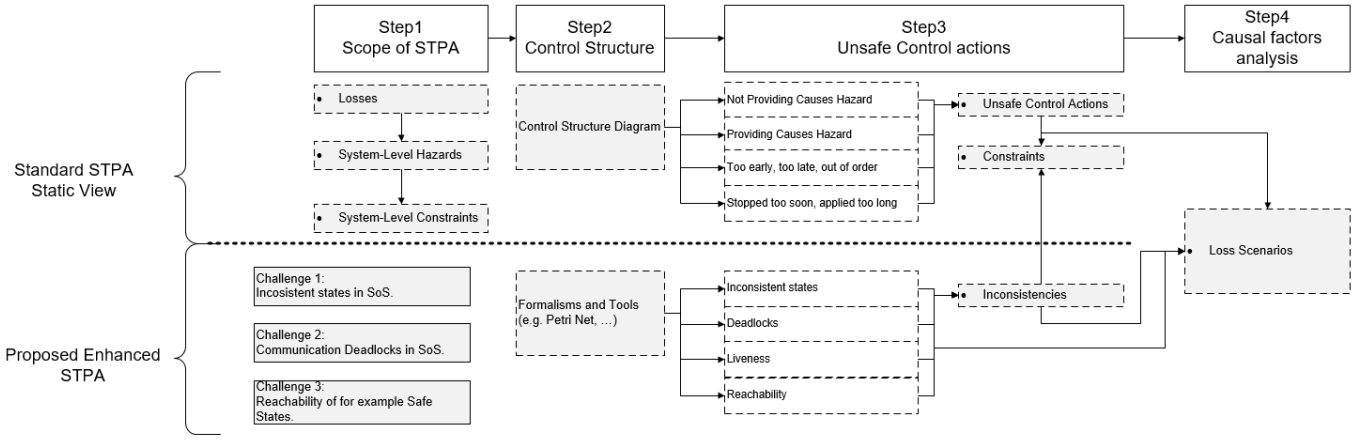

Fig. 4. Adding a dimension to STPA to identify inconsistencies in SoS

lead to critical situations. The states are depicted as Petri net places, while the transitions between states are shown as directed arcs. Specifically, we use timed transitions to simulate timing aspects and communication delays between the involved systems. The process, we propose for analyzing a SoS using Petri nets is as follows:

1) Model the states of each involved system in a separate Petri net. Prepare interfaces to the other systems using open transitions.
2) Connect all Petri nets to one SoS Petri net and adjust the weight of transitions and arcs and place capacity to enable a workflow as intended.
3) Run simulations of the SoS Petri Net to find possible unintended behavior. Adjust even timing of the transitions for different simulations.

B. Step 1 - Modeling single systems in Petri nets:

The states of each involved system are first modeled separately and prepared for later integration. This helps to manage the complexity of the resulting Petri net. We utilize the remote control case described above, but simplify the state machines of the involved systems for the purpose of this paper.

Petri net: HX_01

In Figure 5 the state machine of HX_01 are modeled. While the states of a HX are more complex in reality, we limit our scope and consider only two states, *HX_01 Auto* for showing that the HX is in autonomous mode and controlled by the Fleet Control. The second state of the HX is *HX_01 Remote* when the HX is controlled through the remote control. The transitions we consider here are related to switching between the states as shown in Figure 5 and will be triggered externally.

Petri net: Remote control

Now we model the states of the remote control (Fig. 6). Generally, the remote control can either be connected or disconnected to a specific HX, i.e. *RC Connected* and *RC Disconnected*. Furthermore, the remote control operator can either give the control back to the Fleet Control *Give Control to FC* or request the control of a HX *Request Control*. These states and transitions depend on the overall state of the other systems. The Petri net of the remote control is not modeled as a cycle, because changing states is triggered externally by the Fleet Control server. The interfaces to the Fleet Control server are already provided by using the transitions *Give Control to FC*, triggering that the HX is disconnected from the remote control and *Request Control*, indicating that the remote control requests be connected to a specific HX.

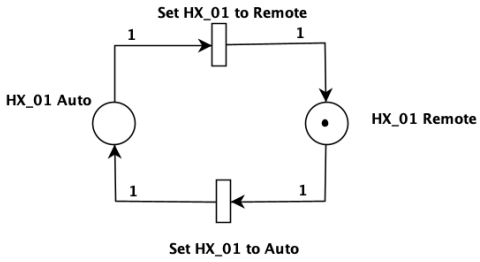

Fig. 5. Petri net for HX_01

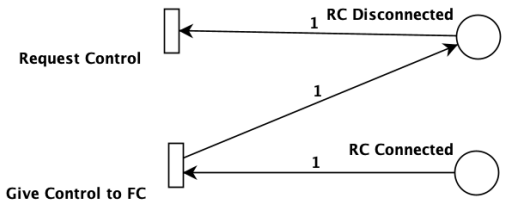

Fig. 6. Petri net for Remote Control

Petri net: Fleet Control

The last system in this context is the Fleet Control. A simplified state machine of Fleet Control is presented in Figure 7 consisting of the states *FC Control HX_01*, indicating that HX_01 is controlled by the Fleet Control server and *FC*

HX_01 Remote, when HX_01 is controlled by the remote control. Once the Fleet Control receives a request from the remote control operator, it can either reject (*Reject Control*) or grant (*Grant Control*) control. If the remote control request is granted, HX_01 shall switch mode to be able to be controlled by the remote control.

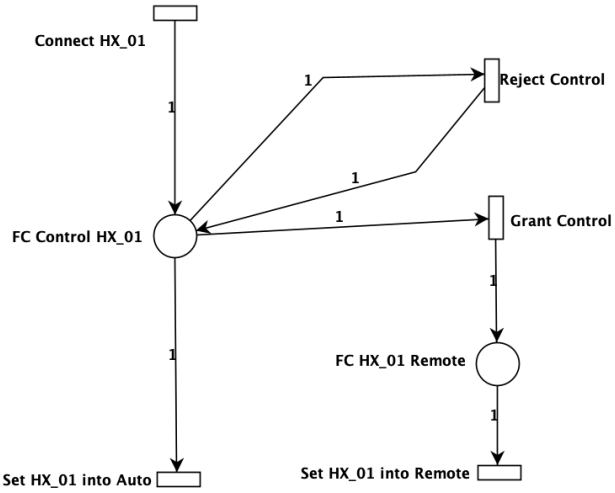

Fig. 7. Petri net for Fleet Control

C. Step 2: Connect all Petri nets to one SoS Petri net

In the separate Petri nets we have used transitions that transform one marker from the input to one marker at an output to the target place. When connecting the derived Petri nets, we generate a larger Petri net as shown in Figure 8 and the weight of transitions and arcs and the capacity of places need to be adjusted to enable the intended workflow.

As a start situation we model HX_01 to be remote controlled shown by the markers in the related places of each involved system. Fleet Control is in the initial state *FC Initial*, HX_01 is in state *HX_01 Remote* and remote control is connected, shown by the marker in state *RC Connected*. Once the Remote control operator is handing over control to Fleet Control, Fleet Control is changing state to *FC Control HX_01*. We consider even the site server in this simulation, where the site operator gets information which HX is connected to Fleet Control. Once Fleet Control is in state *FC Control HX_01*, HX_01 receives a marker, triggering the process in HX_01 moving the marker from *HX_01 Remote* to *HX_01 Auto*. This indicates that HX_01 is controlled by Fleet Control. It is now possible to run a simulation with the resulting Petri net, showing that the workflow is functioning as intended.

D. Step 3 - Run simulations of the workflow

We are interested in situations that are not directly visible and we run the Petri net for a different number of cycles. As one result, we found a deadlock when HX_01 had markers in both states as shown in Figure 9. That HX_01 is in both states at the same time is not realistic, but it might be an indicator for critical controls. This situation needs to be thoroughly analyzed to identify the causal factors and identify possible mitigation strategies. For this case, we found that if the internal states of Fleet Control and the related HX do not match, there is a possibility that a set of different state change requests are send from Fleet Control. Depending on how the HX is managing the incoming state change requests, there is a risk that the HX is set into an incorrect state. A practical example would be, if the remote operator immediately after handing control back to Fleet Control sends a new request to get the control to the same HX. A typical reason could be a human error, when the control is by mistake handed over to Fleet Control. Fleet Control can already be in a state where the command has been sent out to HX_01 to change its state to *HX_01 Auto*. We even tried time delays for communication with HX_01 and we were able to enforce this scenario. The reason for this behavior is the independence of state machines of the involved systems on the one hand and on the other hand possible communication delays.

It is important to identify such scenarios early during development to add additional states, feedback loops for critical messages and safe states where necessary. The identified inconsistencies are documented as shown in Figure 4. In line with STPA, possible design constraints may be derived. In Step 4 of STPA, loss scenarios are analyzed and documented. The found inconsistencies by simulations will provide additional information during the causal factor analysis in Step 4.

VII. Analysis

When analyzing the case from the electric site project we found, that more information is needed, which is not directly visible in the control structure diagrams of STPA. It is important to understand in which state the HX will be once checked out from the Fleet Control. If the HX will be still active, there is a risk, that it will enter an undefined state once disconnected from the Fleet Control. This additional information about system states will be relevant for a hazard analysis. In our model we have only considered one HX to be controlled by Fleet Control or the remote control. The already complex Petri net will become even more complex when more systems and their states are added. Petri nets are useful to identify critical state-related situations in complex system, but the resulting nets can become very complex reducing the maintainability.

A. Limitations

Limitation of Use Case:

The industrial case we have used in this paper is limited to not exceed the scope of this paper. While the states of the HX and Fleet Control are more complex than shown in this paper, we reduced the number of states to highlight how state changes may lead to critical situations. Nonetheless, the complexity of the industrial application makes it even more important to identify inconsistencies of for example the global state.

Correctness of resulting Petri net:

One main challenge with Petri nets we have been facing, is to argue for the correctness of a Petri net. The resulting Petri

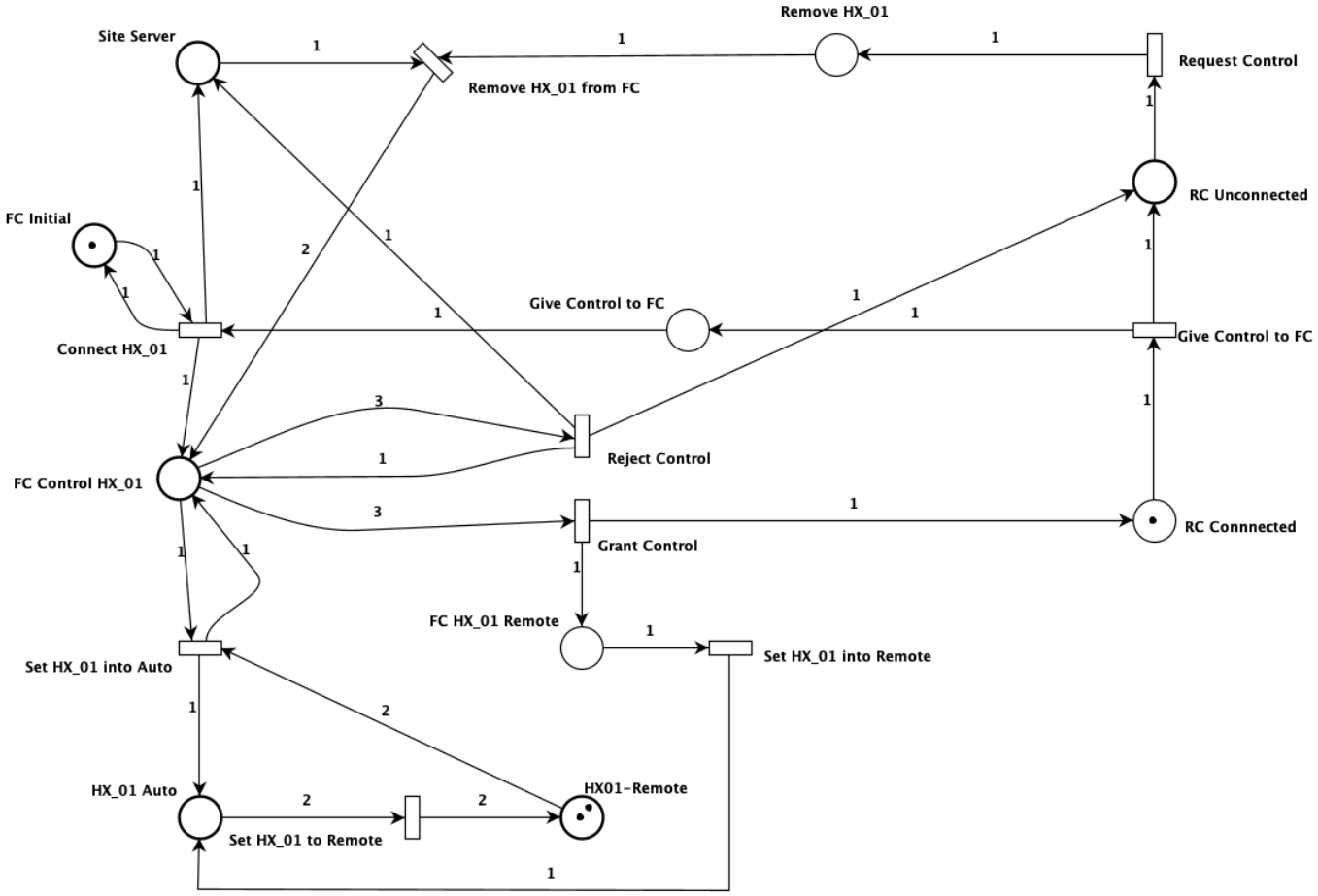

Fig. 8. Petri net for the complete SoS

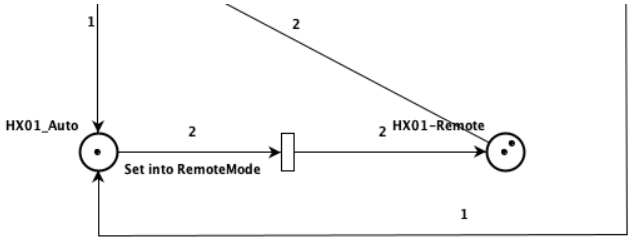

Fig. 9. Petri net for a critical scenario

net, when connecting the Petri nets of the involved systems, needs adjustments in weights of the arcs and transitions as well as adjusting the captivity of places. This might introduce errors in the process flow. We have tried to overcome this challenge by aiming to run a functioning workflow. It is important to choose the right level of abstraction for the Petri net, not to model too many different alternative flows which makes it hard to keep track of targeted workflow.

Complexity of Petri nets:

For large systems with many collaborating subsystems, the associated Petri nets might become very complex and require relevant expertise in building and analyzing them. Model-based engineering methods like SysML are applied by practitioners in industry and SysML state charts can be applied to model the states of the involved systems. Generally, it is possible to transfer SysML state charts to Petri nets as proposed by the authors in [33]. We have not tried this approach to generate the Petri nets presented in this paper. In our work we have shown, that simulation can lead to helpful information for analyzing such critical scenarios, which are not directly derivable from a control structure diagram.

VIII. Conclusion

In this paper we presented a case from the earth moving machinery domain, where autonomous machines are used to transport material in a quarry site. For this system-of-systems, we specifically described a situation where an autonomous machine (HX) is changing from being controlled by a server towards being controlled by remote control operated by a human. We introduced the System Theoretic Process Analysis (STPA), which is a potentially useful approach to analyze the safety for complex systems. STPA is essentially suitable to static monolithic systems and lacks the ability to deal with

emergent and dysfunctional behaviors in the case of SoS. In this paper, we presented an approach for enriching STPA to provide the ability to check whether the distributed constituent systems of a SoS have a consistent perspective of the global state which is necessary to ensure safety. We describe the above approach by taking a specific case of state change related issues that could potentially be missed by STPA by looking at an industrial case. By applying Petri nets we have shown that possible critical situations related to state changes are not identified by STPA. In this context we also proposed a model-based extension to STPA and show how our new process could function in tandem with STPA. This enabled us to simulate the workflow with the goal to find possible flaws in the design. Such information is useful for decision making and development of a SoS.

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 and the SUCCESS Project supported by the Assuring Autonomy International Programme(AAIP), a partnership between Lloyds Register Foundation and the University of York.

REFERENCES

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

[2] ——, "ISO 13849:2015 Safety of machinery - Safety related parts of control systems," 2015.

[3] ——, "ISO 19014:2018 Earth-moving machinery - Functional Safety," 2018.

[4] International Electrotechnical Comission, "IEC 61508:2010 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems," 2010.

[5] C. Ericson, *Hazard analysis techniques for system safety*. Wileys, 2015.

[6] International Electronical Commission, "IEC 61025 - Fault Tree Analysis (FTA)," 2006.

[7] United States Department of Defense, *MIL-STD 1629A - Procedures for Performing a Failure Mode, Effect and Criticality Analysis*, 1980. [Online]. Available: http://www.fmea-fmeca.com/milstd1629.pdf

[8] Volvo Construction Equipment, "Electric Site Project." [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/

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

[10] P. Periorellis and J. E. Dobson, "Organisational failures in dependable collaborative enterprise systems," *Journal of Object Technology*, vol. 1, no. 3, pp. 107–117, 2002.

[11] P. J. Redmond, "A System of Systems Interface Hazard Analysis Technique," Master's thesis, 2007. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a467343.pdf

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

[13] NASA, "NASA Software Safety Guidebook NASA Technical Standard," p. 389, 2004.

[14] International Electronical Commission, "IEC 61882:2001 Hazard and operability studies (HAZOP studies) Application guide," 2001.

[15] ——, "IEC60812:2018 Failure modes and effects analysis (FMEA and FMECA)," 2018.

[16] E. Ruijters and M. Stoelinga, "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools," *Computer Science Review*, vol. 15, pp. 29–62, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2015.03.001

[17] J. L. Peterson, "Petri Nets*," *Computing Surveys*, vol. 9, no. 3, 1977.

[18] W. M. Zuberek, "Timed Petri nets definitions, properties, and applications," *Microelectronics Reliability*, vol. 31, no. 4, pp. 627–644, 1991.

[19] M. A. Marsan, "Stochastic Petri nets: An elementary introduction," 1990, pp. 1–29.

[20] M. El Koursi and P. Ozello, "Using Petri Nets for Safety Analysis of Unmanned Metro System," in *SafeComp 1992*. Elsevier, 1992, pp. 135–139. [Online]. Available: http://dx.doi.org/10.1016/S1474-6670(17)49420-1

[21] D. Weyns, T. Holvoet, and K. Schelfthout, "Decentralized control of automatic guided vehicles: applying multi-agent systems in practice," *Companion to the 23rd*, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1449819

[22] S. Baumgart, J. Fröberg, and S. Punnekkat, "Analyzing hazards in system-of-systems: Described in a quarry site automation context," in *11th Annual IEEE International Systems Conference, SysCon 2017*.

[23] S. Baumgart, J. Froberg, and S. Punnekkat, "Can STPA be used for a System-of-Systems? Experiences from an Automated Quarry Site," in *2018 IEEE International Systems Engineering Symposium (ISSE)*, no. 4. IEEE, 10 2018, pp. 1–8. [Online]. Available: http://www.es.mdh.se/publications/5246-https://ieeexplore.ieee.org/document/8544433/

[24] P. J. Redmond, J. B. Michael, and P. V. Shebalin, "Interface hazard analysis for system of systems," *2008 IEEE International Conference on System of Systems Engineering*, pp. 1–8, 2008.

[25] R. D. Alexander, "Using Simulation for Systems of Systems Hazard Analysis," no. September, 2007.

[26] C. Becker, J. Brewer, L. Yount, D. Arthur, and F. Attioui, "Functional Safety Assessment Of a Generic Electric Power Steering System With Active Steering and Four-Wheel Steering Features," National Highway Traffic Safety Administration - NHTSA, Tech. Rep. August, 2018.

[27] T. Ishimatsu, N. Leveson, J. Thomas, M. Katahira, Y. Miyamoto, and H. Nakao, "Modeling and hazard analysis using STPA," in *European Space Agency, (Special Publication) ESA SP*, 2010.

[28] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer, "STPA-SafeSec: Safety and security analysis for cyber-physical systems," *Journal of Information Security and Applications*, vol. 34, pp. 183–196, 6 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214212616300850

[29] A. Mallya, V. P. B, M. Adedjouma, M. Lawford, and A. Wassyng, *Computer Safety, Reliability, and Security*, ser. Lecture Notes in Computer Science, A. Skavhaug, J. Guiochet, E. Schoitsch, and F. Bitsch, Eds. Cham: Springer International Publishing, 2016, vol. 9923. [Online]. Available: http://link.springer.com/10.1007/978-3-319-45480-1

[30] J. A. Volpe and Van Eikema Hommes, "Assessment of Safety Standards for Automotive Electronic Control Systems (Report No. DOT HS 812 285)," National Highway Traffic Safety Administration, Washington, DC, USA, Tech. Rep. June, 2016. [Online]. Available: https://ntl.bts.gov/lib/59000/59300/59359/812285_ElectronicsReliabilityReport.pdf

[31] J. Zhang, H. Kim, Y. Liu, and M. A. Lundteigen, "Combining system-theoretic process analysis and availability assessment: A subsea case study," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 2019.

[32] D. Zhu, S. Yao, and C. Xu, "STAMP-based hazard analysis for computer-controlled systems using petri nets," *International Journal of Performability Engineering*, vol. 14, no. 9, pp. 1997–2007, 2018.

[33] R. Wang and C. H. Dagli, "An Executable System Architecture Approach to Discrete Events System Modeling Using SysML in Conjunction with Colored Petri Net," in *2008 2nd Annual IEEE Systems Conference*, 4 2008, pp. 1–8.