Combining System-Theoretic Process Analysis and availability assessment: a subsea case study

Juntao Zhang[1], HyungJu Kim[1], Yiliu Liu[1], Mary Ann Lundteigen[1]

[1]Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology NTNU, Trondheim, Norway

Abstract: Hazard identification methods are important tools to verify that the system is able to operate according to specifications under different operating conditions. Unfortunately, many of the traditional methods are not adequate to capture possible dysfunctional behavior of complex systems that involve highly coupled parts, non-linear interactions and software-intensive functionalities. The rather recent method named System-Theoretic Process Analysis (STPA) is one promising candidate to improve the coverage of hazard identification in complex and software-intensive system. Still, there is no guideline for utilizing STPA output to evaluate the potential of loss, which is important for basis for decision-making about system configuration and equipment selection. The focus of this article is placing on the interface between STPA and reliability, availability and maintainability (RAM) analysis. The approach named STPA-RAM model is proposed to translate feedback control loops into Stochastic Petri-nets for discrete event simulation. The proposed approach is demonstrated with a simple case related to subsea design concept. The major conclusion is that STPA-RAM model extends the application of STPA, while also improving, and as such reducing completeness uncertainty and model uncertainty, associated with input data and information for RAM analysis.

Keyword: Reliability, Systematic approach, Complexity, Subsea system

1. INTRODUCTION

Highly coupled parts, non-linear interactions and software-intensive functionalities characterize modern engineering systems. One example could be subsea systems for Oil and Gas (O&G) production and processing. Traditional technologies for subsea control (e.g. hydraulically operated systems) have been gradually replaced by computer-based technology to fulfill the needs of higher level of autonomy, self-diagnostics and monitoring. Such a shift in technologies gives opportunities for more cost-efficient and autonomous operation in marginal subsea fields that have special restrictions associated with accessibility [1]. In this respect, understanding hazards caused by complex interactions on software-intensive systems becomes an important topic. This topic involves two critical steps: the first is to reveal the potential hazards for given design concept, namely hazard identification; the second is to quantify the consequence of critical hazards, to direct engineering efforts to improve reliability, availability and maintainability (RAM) performance.

Subsea control systems include sensors, actuators and controller that interact with the controlled process and other connected systems, such as systems on-board an offshore platform or onshore at the receiving facilities. Loss of critical functionality is not only the result of component faults but also the improper interactions when components are brought together, i.e. the technologies interact in response to the internal and external environment. Unfortunately, identifying hazards arisen from improper interactions is beyond the scope of traditional methods, such as Failure Mode, Effects and Criticality Analysis (FMECA) and Hazard and Operability study (HAZOP) [2, 3]. FMECA focuses on the failure modes and causes of distinct components, whilst HAZOP has a more focus on the consequences of deviations related to process parameters, software functions and procedures. In FMECA or HAZOP, components, process objects, or procedures are analyzed one by one and the interactions are analyzed pairwise. For complex and software-intensive systems, it is important to also complement with analyses that are able to identify failure modes and dysfunctional behavior beyond the physical failures. Some candidate solutions have been proposed by researchers, such as Accimap [4], blended hazard identification method (BLHAZID) [5], functional resonance analysis method (FRAM) [6] and Systems-Theoretic Process Analysis (STPA) [7]. Of the mentioned methods, STPA is the approach attracting the most recent attention due to its suitability to analyze complex and software intensive systems. Some of the advantages and examples of applications of STPA are discussed in [8-10].

STPA is based on a rather new accident causation model named Systems Theoretic Accident Model and Processes (STAMP), which is built on a theoretical basis provided by system theory and control theory [2]. STPA identifies hazards in a systematic way by examining the potential deviations on the defined feedback control loop. A feedback control loop is a graphical representation, which involves all the actors that have impacts on the emergent system properties in form of their individual behavior and interactions. Each actor is identified by its responsibilities (e.g. tasks/commands) and its reliance on information/feedback. The improper or inadequate combination of control commands and feedbacks can result in loss of vital values, such as human losses, environmental losses, customer dissatisfaction and economic losses. STPA has been applied in different applications such as automotive [11], healthcare [12], aerospace [13], maritime [14] and subsea [9, 15]. As a hazard identification method, STPA can be naturally embedded in safety and security analysis [16, 17] by guiding the associated controls and mitigating measures depending on different applications [12, 18, 19]. So far, the commonality and acceptance of STPA are limited to the academic studies and not yet adapted as best practices in e.g. international standards on safety assessments. Yet, it seems very promising to use STPA as complementary to FMECA and HAZOP to efficiently increase the coverage of hazard identification thus reduce the potential of accidents [10].

STPA provides an alternative model to identify hazards of complex and software-intensive systems. Yet, STPA has no interface with RAM models thus it is not fully clear how to interpret STPA outputs in the decision context. RAM models characterize the combinations of evolutions (e.g. degradation and failure) and maintenances (e.g. replacement and repair), and is used to demonstrate a certain level of RAM performance before the new design concepts for systems are qualified for the intended use. Few attempts have been made to systematically use STPA outputs to improve RAM models, whereas a similar link can be readily found for traditional methods, e.g. FMECA and HAZOP. The lack of this connection is unfortunate as important insight can be overlooked and not transferred from STPA to RAM model. This is also pointed out by Hafver et al. [20], who suggest that the STPA output has the potential to construct better RAM model to predict the effect of improper/inadequate controls on system behavior.

With regard to the nature of modelling, RAM models can be classified as combinatorial approaches, or state-transition approaches that rely on event-chain description. Fault tree analysis (FTA) is one example of combinatorial approaches, where the occurrence or probability of loss is determined directly by the combination of events related to equipment failure and indirectly by the impact of human factors and external events [21]. Such a combinatorial approach holds strict assumption on independence between events, so it is only able to cover accident related to hardware that fails as a chain of event. The classical state-transition approaches are Markovian approach [22-24] and Stochastic Petri-nets (SPN) [25, 26], which prove to be more efficient in reflecting dynamics features of system behavior than combinatorial approach by paying the price of calculability [22, 27]. STPA has been able to identify dependencies with loss consequences beyond what is normally captured by FMECA and HAZOP. It is therefore of interest to investigate how STPA results can be utilized for constructing state-transition models. In this article, SPN is selected as it is theoretically more expressive than Markovian approach in terms of event synchronization and flow propagation [28]. From literature, some initial proposals to combine STPA with SPN have been found. They are mainly for qualitative analysis, for example to derive integrated hazard logs for safety-guided design [29] and to have formal models for conducting STPA [30]. Yet, adding quantitative analysis in STPA has not been fully exploited.

The main objective of this article is therefore to propose a new model named STPA-RAM to supplement qualitative STPA with quantification models using SPN. STPA is conducted to identify hazardous scenarios, by modelling system behavior into feedback control loops. The perturbation initiated on controller or controlled process can propagate into system-level losses if no constraint is enforced to invert the condition of having hazard. Considering the feedback control loop obtained in STPA is not an executable model, SPN is to model the coordination between the controlled process and controller, and simulate the system response under specified variations of feedback control loop thus predict frequency of losses. An illustrative case study is carried out to demonstrate the application of proposed approach.

The following section 2 introduces original STPA succinctly and give some reflections about its applications. Section 3 proposes a step-wise approach for building STPA-RAM model, and describes how to structure feedback control loops into SPN. A conceptual subsea design is selected to illustrate the application of proposal in section 4. Finally, discussions and concluding remarks are presented in section 5.

2. STPA PROCEDURE AND APPLICATION

2.1 Overview of STPA procedure

The STPA approach has been under continuous development since emergence, and its framework can be complicated with respect to the analytical needs and constraints for practical use, e.g. [31]. This article follows the generic steps suggested in STPA handbook by Leveson and Thomas [7], which are illustrated in Figure 1:

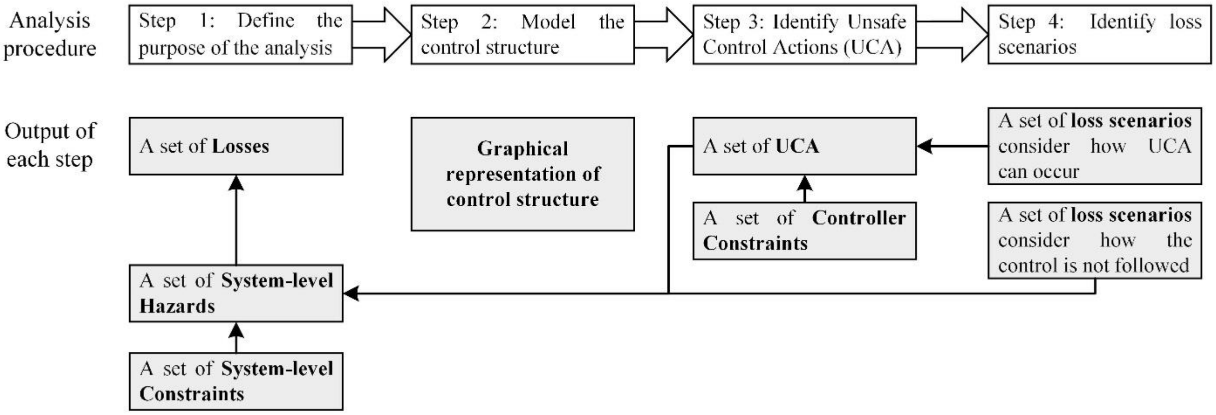

Analysis procedure
Step 1: Define the purpose of the analysis
Step 2: Model the control structure
Step 3: Identify Unsafe Control Actions (UCA)
Step 4: Identify loss scenarios
Output of each step
A set of Losses
Graphical representation of control structure
A set of UCA
A set of loss scenarios consider how UCA can occur
A set of Controller Constraints
A set of loss scenarios consider how the control is not followed
A set of System-level Hazards
A set of System-level Constraints

Figure 1 the framework of STPA and its output

- **Step 1: Define the purpose of analysis.** The first step is to define the scope of analysis by identifying the consequences on system level in presence of any single or multiple variations on feedback control loop. The consequence includes the losses and associated hazards. *Losses* could be any type of dissatisfactory value to stakeholder when the system fails to achieve its goal and objective, and system-level *hazards* are a set of system states that can lead to losses together with worst-case conditions. Such broad definition of losses and hazards implies that STPA covers traditional safety issues as well as RAM issues.

- **Step 2: Model the control structure.** The next step is to develop feedback control loops. The hierarchical control structure is composed into one or more feedback control loops, and visualize actors involved, control actions and feedback information. The objective is to have the global and complete vision about the hierarchy concern being controlled, thus supports the following step 3 and step 4. An example of a feedback control loop is illustrated in Figure 2, from the left to right the details are added based on the responsibilities assigned to each actor. The hierarchical control structure can be refined until the suitable granularity is reached.

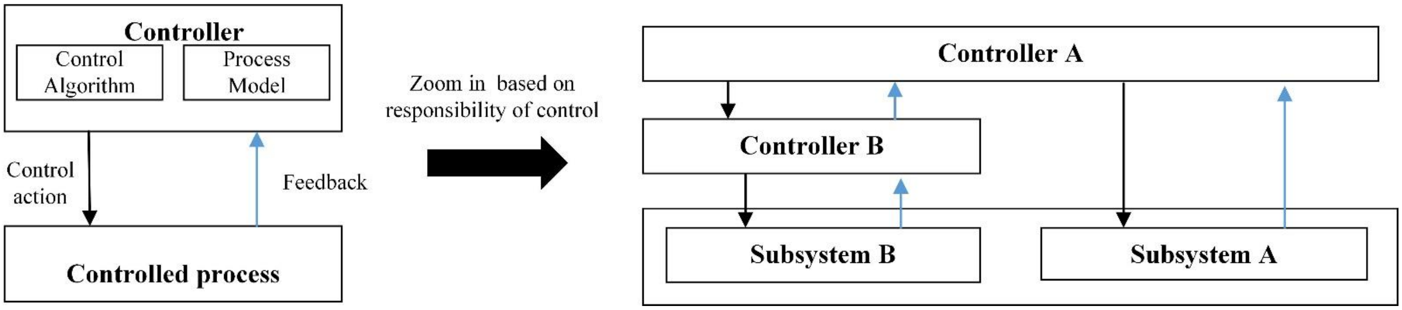


Figure 2 Example feedback control loop

- **Step 3: Identify Unsafe Control Actions (UCAs).** The third step relies on the structured identification of what can go wrong, using the feedback control loop and a prepared context table as basis. The output of this step is a list of UCA that in particular context results in one or more of the hazards identified in step 1. The UCAs are be identified through four guide conditions taking advantage of control structure: (1) the control action is not provided, (2) the unsafe control action is provided, (3) control action is provided too late, too early, or out of sequence and (4) control action is stopped too soon or applied too long (applied only for continuous control). The constraints for controller can be defined as conditions or behaviors to prevent occurrence of UCAs (and ultimately prevent related hazards).
- **Step 4: Identify loss scenarios.** Loss scenarios are used to describe the casual factors that lead to hazards (and ultimately to losses in worst-condition). The first type of loss scenarios consider how the UCA can occur, including the causes of unsafe controller behavior and inadequate feedback. The second type of loss scenario consider how the safe control action is not followed, including the causes of deviated control path and controlled process. The control structure obtained through step 2 need further refinement by including the sensors and actuator of the control loops so that analysts can examine why the feedback is not detected or wrongly detected and why the control action is not followed or improperly followed by actuators.

The new insight brought by STPA is the characterization of erroneous or inappropriate control and associated causality knowledge. All the possible contributions to the losses of a system (i.e. hardware, software, human and organizational factors) are considered as the elements (i.e. controller and controlled process) in the feedback control loop. The loss scenarios are determined when the *combination of* control commands, inadequate feedback, and the state of the controlled process and its environment is inadequate or improper. Such systematic way of hazard identification goes beyond the scope of traditional methods based on the common engineering sense (i.e. hardware-wise). In this respect, we argue that STPA is suitable for analyzing subsea system built today, which becomes increasingly intelligent and more dependent on software.

Whereas STPA theoretically increases the coverage of hazards, the current framework of STPA strictly emphasizes on qualitative aspects and has no guidance on how to direct the further quantification. In such set-up, STPA has no guidance on how to direct the further quantification of loss scenarios, which leaves designers with challenging tasks to interpret STPA results in the decision making. The architect of STPA, Leveson [2] has argued that quantitative analysis in STPA is questionable, for mainly two reasons. First, pursuing quantitative analysis can distract the attention away from important causal factors that are not characterized statistically [32]. Second, it requires probabilistic insights about future events that are not supported by historical data. Assigning probabilistic information for loss scenarios is a challenging and error-prone task even with excessive elaborations among system designers and experts.

Yet, there are also some reasons to extend STPA on a more quantitative basis. First, it is hardly possible to eliminate all possible loss scenarios in reality as countermeasures may degrade or become less efficient over time, see examples in [8, 19] where STPA is applied to technical system. It is therefore necessary to evaluate the effect of loss scenarios versus considerable costs for provision of countermeasures. Second, the lack of data for probabilistic model does not mean the probabilistic model is useless in the context of STPA. The similar problem has been discussed by Bjerga et al. [33], who argued that rather than being pessimistic to discard probability, it is needed to advocate probabilistic analysis to address risks induced by potential systemic accidents, so that STPA results can be confidently used in a decision context.

In a short summary, we argue that current STPA framework has both advantages and inadequacies. Although STPA reveals a full spectrum of vulnerable points for given design concept, it leaves all judgments about prioritization of design improvements and modifications to the designers. The effect of designed countermeasures may not be obvious without constructing quantification model. Stimulating how the system responses to perturbations on feedback control loop through a defined mathematical framework can be a solution to this problem. That is the topic of next section.

2.2 Theoretical basis for simulation

According to Thomas [34], an UCA (and its descendant – loss scenarios) can be defined with a formal structure as a quadruple <***Ac***, ***CA***, ***Co, U***>, where:

- ***Ac*** is a set of actors refer to at least one controller of the controlled process.
- ***CA*** is a set of control commands issued by controller Ac ∈ ***Ac***.
- ***Co*** is a set of contexts that defines a unique system state, which implies whether the control action is needed (given) or not. ***Co*** can be specified explicitly or implicitly in terms of distinct variables. Each Co for the controller Ac should be independent.
- ***U*** is a set of hazardous state (i.e. description of possible and relevant losses). To be qualified as UCA, a control action must satisfy the property that (Ac, CA, Co) can lead to at least one of U ∈ ***U***

A control process can be equivalently transferred into Finite State Automata (FSA). FSA is used to model the discrete behavior of system, consists of a finite number of state, transitions between states and events. The *state* represents a quiescent node in the sequence of a control process, and the *event* describes the control action to be performed. A control-like *transition* triggered by an event or condition can cause the change of state. For instance, if providing a control action under a specific context that causes hazards, the transition function is $\boldsymbol{T}$: $\boldsymbol{Co} \times \boldsymbol{CA} \rightarrow \boldsymbol{U}$. In this sense, the system in question is reformulated as the closed-loop control where the feedback signals (i.e. state of system) are now being used to both control and adjust itself.

The change of states (i.e. ***Co***) is modelled by random and deterministic events defined for a system. RAM model is one example, in which the failure and degradation are considered as stochastic events and software updates and hardware replacement are considered as deterministic. Therefore, one can establish the interface between RAM model and loss scenarios derived by STPA through FSA. The effect of loss scenarios on RAM performance can be simulated by FSA under the following assumptions: The transitions between states describe the situation where the control actions (no matter safe or unsafe) update values of model parameters (e.g. failure rate) in the new state. The changes made for model parameters influence the related transitions in FSA as a function of time. For example, a shutdown valve may be exposed to the hard stress in the situation of ‘slam shut’ closure, which can be seen as a loss scenario and its consequence is the permanent damage on valve. This implies the accelerated degradation rate for the shutdown valve once reaching the hazardous state that defines above situation.

Given such settings, the next chapter presents the proposal for hazard quantification, named STPA-RAM model. SPN is selected as the suitable modelling approach that follows state-event transition formalism.

3. PROPOSAL: STPA-RAM MODEL

3.1 Two-step approach

Figure 3 illustrates the two-step approach: The first step is to carry out an STPA to identify loss scenarios. The second step has to main sub-tasks: (i) to prepare RAM model using available specifications for the system and its intended functions, and (ii) to complement this model with new information from STPA in the first step. The outcome is a revised RAM model representing new information about dependencies in the feedback control loop developed by the STPA, namely a STPA-RAM model.

In the approach, the STPA-RAM model can reflect the potential deviations in different feedback control loops and interfaces between feedback control loops. Causality knowledge obtained in STPA is maintained in the STPA-RAM model. The loss scenarios can be generated by studying the reachability to the hazardous states. The actors of feedback control loops (i.e. hardware, software and organizational factors including human) are closely tied together in FSA in which the interdependencies between feedback control loops are represented by transitions. To maintain in the same format for integration, RAM model is constructed as the feedback control loop. In this regard, the monitoring and inspection on the state of controlled process are the considered as the feedback loop to the maintenance and intervention controller, whose responsibility is to update the software or replace the hardware when the feedback indicates the malfunctions and deviations of controlled process. Such modelling approach goes beyond the classical RAM model that is built on propagating the information from low-level system hierarchy along with simple logics.

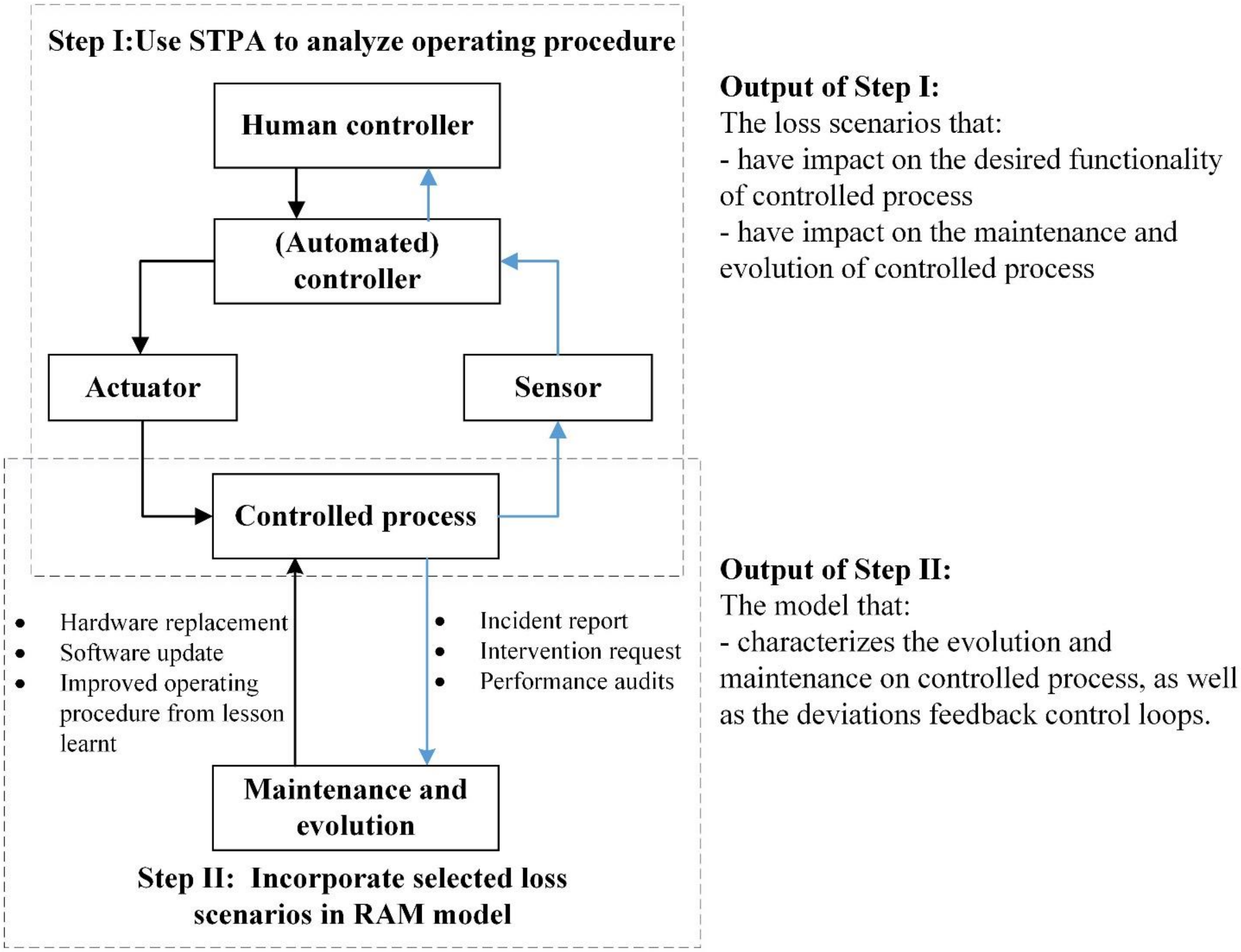

Step I:Use STPA to analyze operating procedure
Human controller
(Automated) controller
Actuator
Sensor
Controlled process
Hardware replacement
Software update
Improved operating procedure from lesson learnt
Incident report
Intervention request
Performance audits
Maintenance and evolution
Step II: Incorporate selected loss scenarios in RAM model
Output of Step I:
The loss scenarios that:
- have impact on the desired functionality of controlled process
- have impact on the maintenance and evolution of controlled process
Output of Step II:
The model that:
- characterizes the evolution and maintenance on controlled process, as well as the deviations feedback control loops.

Figure 3 Two-step approach for STPA-RAM model

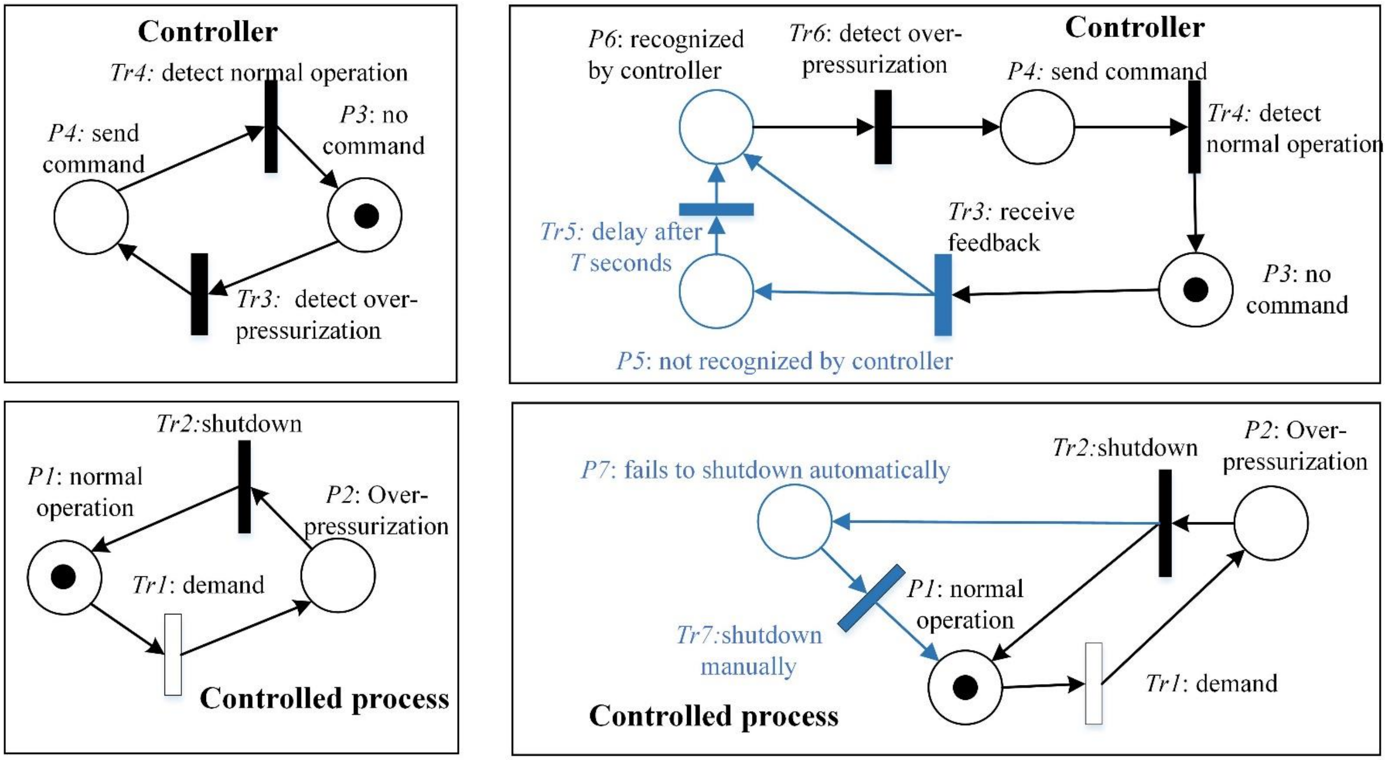
The proposed approach covers multiple models and the coordination between models are rather complex. The complexity here depends on the number of feedback control loops. The original feedback control loop defined in STPA is inadequate to express such complex coordination and has no execution ability. SPN that follow the state-event transition formalism is selected to structure models of proposed approach, without distorting the feedback control phenomenon of STPA. It may be noted that SPN is only one of many ways to visualize such interactions and construct the executable model. The other methods obeying state-transition formalism can achieve the same objective but they are not further discussed in this article.

3.2 Use SPN to construct STPA-RAM model

The SPN model consists of a *net structure* and a *marking* [35]. The net structure is made of the *places* (represented by circles), *transitions* (represented by bars), and their connection (presented by directed arcs). The arc links a place to a transition is called input arc and the arc links a transition to a place is called output arc, and they can be assigned with a natural number, named *weight* or *multiplicity* (normally assumed to be 1). Places may contain *tokens* (represented by bullet), which can move between places when enabled transition is fired. The transition is enabled when a number of token on each of its upstream places (a place connected by input arc) is not less than the weight/multiplicity of input arc. The transition is fired when the associated delay elapses (given that transition remain enabled during delays). The time delay between enabled transition and firing can be characterized as fixed or random [26]. The marking represents the distribution of tokens on a net structure. In such setting, the place of SPN can specify the context as premise condition for control action, and the tokens specify the state/value of context that decides whether the control action is needed or not. The transitions represent the control actions and information feedback on feedback control loop, and the time-dimension of control process is introduced by the random or fixed delays. In addition, *predicates* and *assertions* by means of variables can be introduced to SPN [36]. Predicate (often represented by '?') is a formula to validate/disable the transitions when variables are verified/unverified, and assertion (often represented by '!') is a formula to update the variables after the associated transition is fired. The predicates can model synchronization between control actions and controlled process, and the assertion is used to capture the transformational change in the system as the result of executed control actions. The detailed information about how to construct SPN model can be found in [28, 36]. The rest of this section introduces a small example for using SPN to construct STPA-RAM model.

Figure 4 illustrates a generic feedback control loop represented by SPN model. Two piecewise SPN models are structured to represent the behavior of controller and controlled process. The controlled process (i.e. system) can become abnormal and this is assumed as a stochastic process. The responsibility of controller is to intervene with the controlled process when it is in abnormal state, and system state is either maintained or, when relevant, reset to normal within the permitted time (*X* seconds). The two variables considered for predicates and assertions here are denoted as *normal_state* and *reset*.

Figure 4 (a) illustrates SPN model for the defined feedback control loop, assuming there is no loss scenario as the result of adequate control. The tokens initially stay in *P1* and *P3*, representing the state that the system is normal so no need to intervene the system. The initial marking is that one token stays in *P1* and one token stays in *P3*, indicating that normal state of system and no control command. When the token reaches *P2* from *P1* after firing the transition *Tr1* (i.e. system state becomes abnormal), the assertion of *Tr1* is '! *normal_state* =false'. Then, the transition *Tr3* is fired as the predicate of *Tr3* is '? *normal_state* =false', means that the controller sends the command to activate the system when abnormal state is detected (by controller). Similarly, when the token reaches *P4* through transition *Tr3*, the variable *reset* is assigned as *true* to fire the transition *Tr2* (i.e. send command to reset the system/controlled process). When the token leaves from *P2* to *P1* (means the activate process is completed after certain delay), the variable *normal_state* is updated as true so that transition *Tr4* can be fired.



(a) (b)

Figure 4 Mapping control structure into SPN: (a) adequate control and (b) two potential loss scenarios

Figure 4 (b) illustrates how we suggest modeling the influence of STPA output in SPN where two loss scenarios have been selected, and they are represented by net structure colored as blue. Loss scenario 1 is that controller sends the command too late (after *T* seconds) when abnormal state is detected, which leads to the hazard denoted as H.1. In this case, the transition *Tr3* in Figure 4 (a) is divided to two transitions *Tr3* and *Tr6* in Figure 4 (b) to distinguish between the event ‘*receive feedback of state*’ and the event ‘*abnormal system state has been recognized (by controller)*’. In addition, two new places *P5* and *P6* are introduced to represent the context that ‘*feedback has been recognized too late*’ and ‘*feedback has been recognized immediately*’ respectively. The loss for H.1 is expressed as the extra *T* seconds that system is exposed to the abnormal state, equals to the delay of transition *Tr5*. Loss scenario 2 is that system is not successfully activated in response to the command and that a manual reset (intended to compensate) leads to hazard denoted as H.2. In this case, the transition *Tr2* in Figure 4 (a) is divided to two transitions *Tr2* and *Tr7* in Figure 4 (b) to distinguish between the event ‘*reset system upon control command*’ and the event ‘*reset system manually*’. The new place *P7* is introduced to represent the state that ‘*the system fails reset automatically*’. The associated loss for H.2 is that the system is exposed to more stress when it is manually activated then the system is more prone to be abnormal in the rest of operation, saying that the transition rate of *Tr1* is slightly increased by α% after the transition of *Tr7*. The transition *Tr2* now has two downstream places: *P7* and *P1*. The frequency of loss scenario 2 can be denoted as the probability that token from *P2* enters into *P7* when transition *Tr2* is validated, that is ‘? *reset* =true’. Similarly, the frequency of loss scenario 1 can be denoted as the probability that token from *P3* enters into *P5* when transition *Tr3* is validated, that is ‘? *normal_state* =false’. Table 1 summarizes the synchronized product for Figure 4 (b).

Table 1 Synchronized product of case in Figure 4 (b)

Transition	Predicate	Assentation	Delay of transition
Tr1		normal_state=false	Stochastic delay, λ
Tr2	reset =true	normal_state =true	*X* seconds
Tr3	normal_state =false		0
Tr4	normal_state =true	reset =false	0
Tr5			*T* seconds
Tr6		reset =true	0
Tr7		$\lambda = \lambda \times (1+ \alpha)$	0

Although a quite simple and restrictive feedback control loop is considered in Figure 4, the above example is sufficient to illustrate how to construct STPA-RAM model by SPN. One specific issue is the refinement of SPN. The SPN in Figure 4 could be further refined by including SPN that represent sensor and actuator in the same feedback control loop or other actors from different feedback control loops. The coordination between actors are realized by the variables that are updated by assertion and propagated in feedback control loop by predicates. For instance, if the controller wrongly believes that the system is in abnormal state, a possible cause can be that the sensor provides the wrong feedback of actual state of system. To model this casual factor, one may construct another piecewise SPN that represent the evolution of sensor performance, e.g. *state_sensor*. The predicate of transition *Tr3* is subjected to the variable *normal_state* and *state_sensor*. The detailed example is given in the case study that follows in the next chapter.

4. CASE STUDY

In this section, the proposed approach is applied on a novel design concept of subsea architecture named Subsea Gate Box (SGB) that arises in Subsea Production and Processing SUBPRO [37] research center. The detailed introduction to this design concept can be found in [38]. Some simplifications are made on the original design concept for illustrative purpose. The modelling and simulation of SPN is completed by the software GRaphical Interface for reliability Forecasting (GRIF) [39] with the simulation engine Moca-RP.

4.1 System description

SGB is new field architecture concept where it is possible to install dedicated solutions for each well or a group of well considering the particular needs of subsea processing, i.e. boosting, metering and separation. The advantages of this design concept are in form of increasing oil and gas recovery, operation flexibility of separation and process efficiency. Figure 5 presents one alternative configuration for SGB, where each SGB consists of three functional modules: separation module (SPM), choke valve module (CVM) and multiphase pump (MPM) module. The normal processing line consists of SPM and MPM, where hydrocarbon flow is separated by SPM into liquid and gas, where the liquid is pumped through multiphase pump and the gas is assumed to flow naturally to the manifold. When the functional modules of the normal processing are faulty, the hydrocarbon can be bypassed to CVM on the same SGB. The choke valve then controls hydrocarbon pressure with low production efficiency. A subsea control system that interacts with the SGB equipment and sensors is vital for maintaining an optimal operation. The switch between processing lines is controlled by subsea controller (s) and realized by the open/close of crossover valve (XOV). SPM, MPM and CVM are retrievable. The connection between module (e.g. isolation valves and pipe connectors) and the sensors (e.g. transmitters of flow, temperature and pressure) are not illustrated in Figure 5.

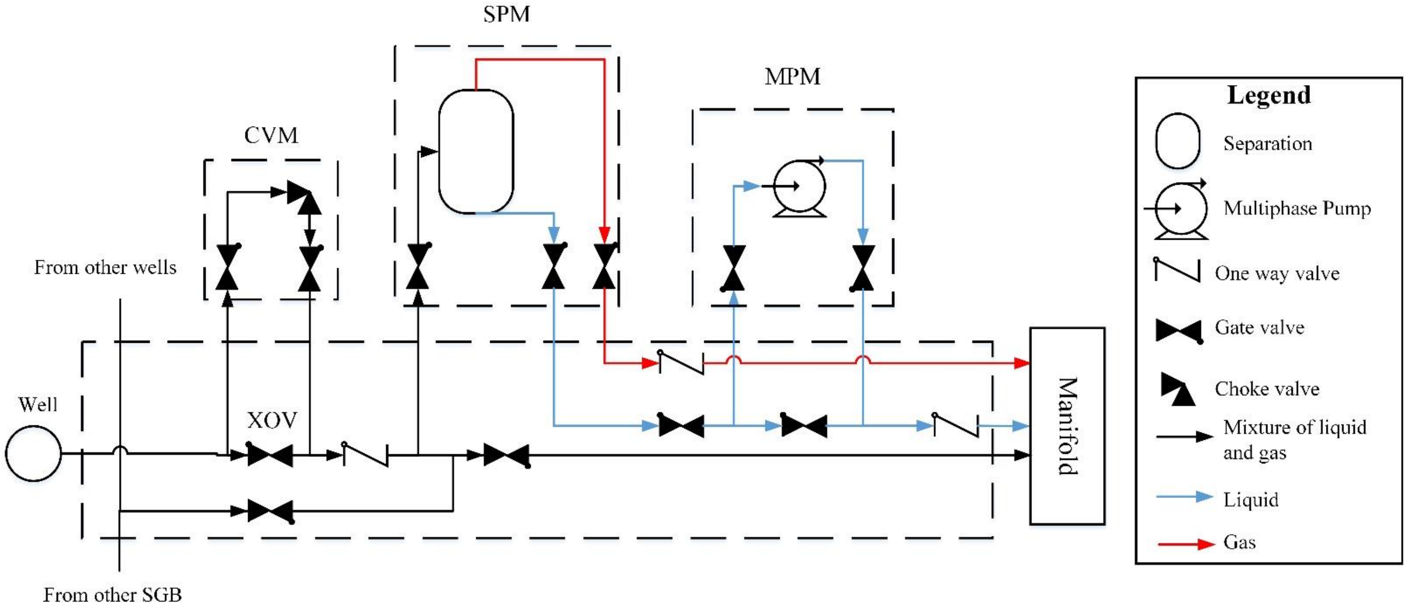


Figure 5 System schematic drawing of SGB

In the following subsections, the STPA-RAM model is constructed for illustrative purpose. The first step is to carry out STPA for analyzing the operating procedure of SGB. The involved actors for the control action are simplified as normal processing line (SGB-NP), bypass processing line (SGB-BP), XOV, sensor and controller. The second step is to build up RAM model considering the state of actors. Some data inputs for RAM models are assumed for demonstrating the approach only. Given the numerical results obtained through the STPA-RAM model, the countermeasures for selected loss scenarios are suggested. The selection of countermeasures are not discussed as the cost information for suggested measures are not available currently.

4.2 Step I: Carry out a general STPA

Based on the discussion with the system designer, three types of losses were identified: unexpected decrease in production efficiency (L.1), hydrocarbon spills (L.2), and complete shutdown of SGB (L.3). The associated system level hazards and associated constraints are summarized in Table 2.

Table 2 System-level hazards and constraints for SGB

System level hazard (SH)	**System-level constraints (SC)**
SH.1: Hydrocarbons flow into non-optimal processing line [L.1]	SC.1 Hydrocarbons must always flow into optimal processing line
SH.2: Hydrocarbons flow into unavailable processing line [L.1, L.2, L.3]	SC.2 Hydrocarbons must never flow into unavailable processing line
SH.3: Over-pressurization of equipment in selected processing line [L.2, L.3]	SC.3 Pressure must never be built-up above design limit

The high-level hierarchical control structure is illustrated in Figure 6. The subsea controller consists of process control system (PCS), subsea control unit (SCU), process shutdown system (PSD), subsea control module (SCM) and subsea electronic module (SEM). The structure and complexity of subsea controller depend on the operating strategies and distance to controlled equipment [15]. For instance, PCS and PSD located on surface facility deliver the command from human operator to control equipment and shut down the system, through SCU to the SCM/SEM that located subsea. To simplify the case study, only SCM and SEM are considered, and the responsibility is distribute the control commands to equipment. When the ability to use the normal processing line is lost, human operator sends the coded command to SCM/SEM that distributes the command to associated valves. The SGB-NP is shut down by the closure of isolation valve, and XOV is opened thus the hydrocarbon is redirected to CVM with lower production efficiency. When the normal processing line is restored after maintenance, then human operator sends the command through the similar process to restart SPM and MPM and redirect flow to normal processing line.

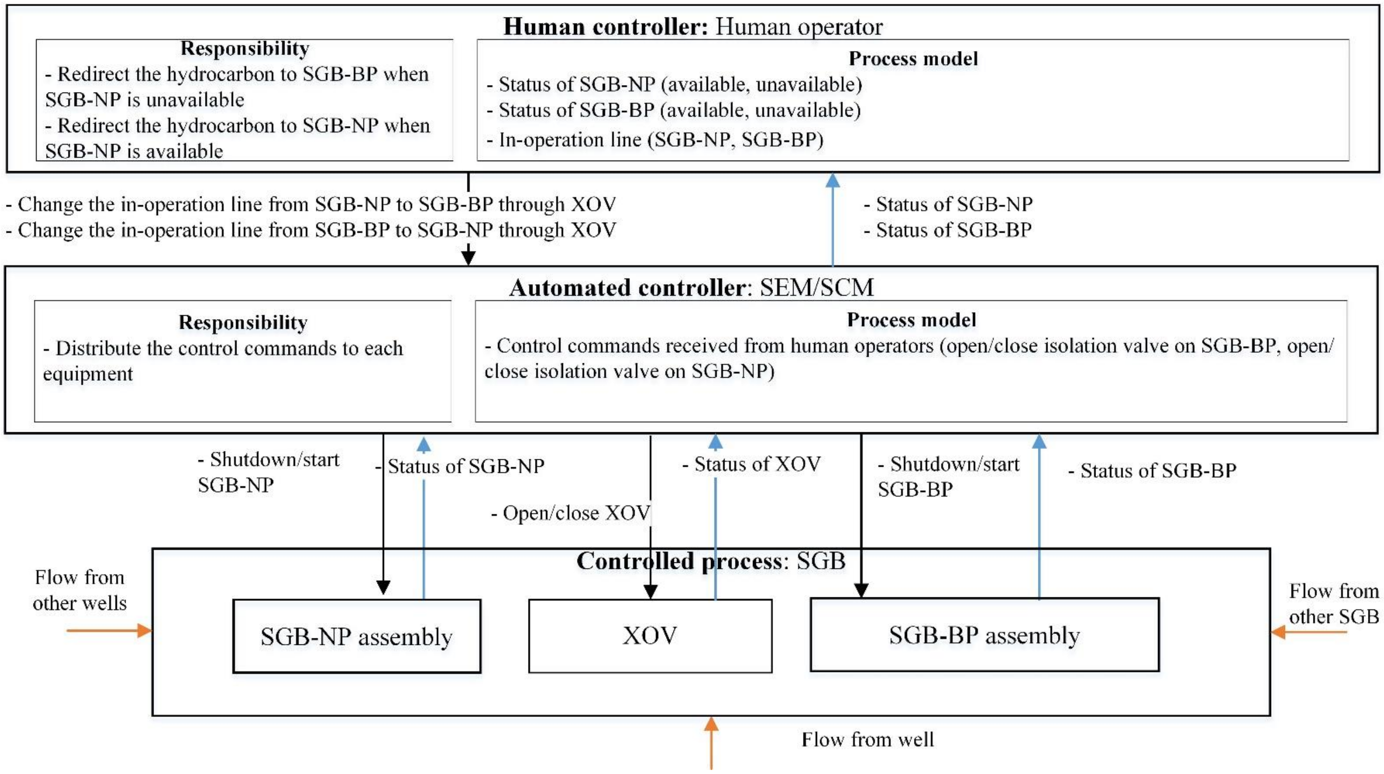

Human controller: Human operator
Responsibility
- Redirect the hydrocarbon to SGB-BP when SGB-NP is unavailable
- Redirect the hydrocarbon to SGB-NP when SGB-NP is available
Process model
- Status of SGB-NP (available, unavailable)
- Status of SGB-BP (available, unavailable)
- In-operation line (SGB-NP, SGB-BP)
- Change the in-operation line from SGB-NP to SGB-BP through XOV
- Change the in-operation line from SGB-BP to SGB-NP through XOV
- Status of SGB-NP
- Status of SGB-BP
Automated controller: SEM/SCM
Responsibility
- Distribute the control commands to each equipment
Process model
- Control commands received from human operators (open/close isolation valve on SGB-BP, open/close isolation valve on SGB-NP)
- Shutdown/start SGB-NP
- Status of SGB-NP
- Open/close XOV
- Status of XOV
- Shutdown/start SGB-BP
- Status of SGB-BP
Controlled process: SGB
Flow from other wells
SGB-NP assembly
XOV
SGB-BP assembly
Flow from other SGB
Flow from well

Figure 6 High-level control structure of SGB

On the basis of control structure defined above, we identified UCAs. Some examples are reported in Table 3. The loss scenarios (SO) can be further identified (here using UCA.1 as example) as reported in Table 4. In addition, Table 5 identifies the loss scenarios related to the situation that human operator sends the correct control command to change from SGB-NP to SGB-BP but it is not followed or improperly followed by automated controller. It is assumed that some suggested countermeasures in Table 4 and Table 5 have been derived from analyses carried out for the purpose of this article. It is expected that more detailed analysis with improved results would come with an updated analysis when the SGB has reached a more mature design stage.

Table 3 UCAs for defined control structure

Control action from SEM/SCM	**Identification of UCAs**			
Change the in-operation line from SGB-NP to SGB-BP through XOV	Not provided	Provided	Wrong timing or order	Too soon or too long
	UCA.1: Control command is not provided when SGB-NP is faulty and XOV is available [SH.1, SH.2,]	UCA.2: Control command is provided when both SGB-NP and XOV are available [SH.1]	UCA.4: Control command is provided too late when SGB-NP is faulty and XOV is available [SH.2, SH.3]	UCA.5: Control command is stopped too soon before XOV is fully closed when SGB-NP is faulty [SH.2, SH.3]
		UCA.3: Control command is provided when both SGB-NP and SGB-BP are faulty [SH.1, SH.2]		

Table 4 Loss scenarios related to UCA.1 and suggested countermeasures

<table>
<tr><td colspan="2">UCA.1: Change the in-operation line from SGB-NP to SGB-BP through XOV is not provided by SCM/SEM on command from human operator when SGB-NP is faulty and XOV is available [SH.1, SH.2]</td></tr>
<tr><th>Loss scenarios</th><th>Suggested countermeasures</th></tr>
<tr><td>SO.1 for UCA.1: Human operator receives correct feedback but interprets it incorrectly so SEM/SCM does not receive control command from human operator. The causal factor is that human operator lacks sufficient understanding for abnormal situation.</td><td>Must provide the sufficient training for operators to deal with specified hazardous situations.</td></tr>
<tr><td>SO.2 for UCA.1: Human operator receives correct feedback but makes mistakes so SEM/SCM does not receive control command from human operator. The causal factor is that human operator is overstressed when there are too many process to be considered.</td><td>The reference document must be presented to provide guidance for operation.</td></tr>
<tr><td>SO.3 for UCA.1: Human operator receives incorrect feedback about conditions of SGB-NP so wrongly believes that the SGB-NP is working but it is not. The casual factor is that the sensor on SGB-NP provides erratic readings.</td><td>Sensors must be monitored continuously and be calibrated when erratic reading was detected</td></tr>
</table>

Table 5 Detailed loss scenarios and suggested countermeasures

Loss scenarios	**Suggested countermeasures**
SO.4: The control command is initiated by human operator but not received by SCM/SEM. The casual factor is that there is a critical failure on SEM/SCM [SH.1, SH.2].	The status of SCM/SEM must be checked before operation and after each updates.
SO.5: The control command is provided by SCM/SEM on command from human operator, but actuator does not responds to this control command. The casual factor is critical failures on XOV (actuator) [SH.1, SH.2].	XOV must be checked regularly and be repaired when critical failure is revealed.

The suggested countermeasures may degrade or become less efficient considering operating conditions of SGB. For instance, the availability of XOV cannot be guaranteed by continuously monitoring and repair due to maintenance in subsea context may be delayed considering the availability of vessel that transport spare parts. In addition, the cost of some suggested countermeasures may be considerable. For instance, monitoring potential faults in sensor measurements often requires a reference sensor to be installed with additional costs for purchasing and installation. Therefore, designers would like to evaluate the cost-benefit before selecting countermeasures. In this case study, two loss scenarios that caused by erratic reading on sensors are investigated to exemplify:

- Loss scenario 1 (LSO1): Human operator receives incorrect feedback about conditions of SGB-NP due to erratic readings of sensor and wrongly believes that the SGB-NP is faulty but it is not. The control command to stop SGB-NP and activate SGB-BP is provided accidently (SH.1). It is assumed that this situation is recognized after 360 hours and the system operates in reduced production efficiency during this period (L.1).

- Loss scenario 2 (LSO2): Human operator receives incorrect feedback about conditions of SGB-NP due to erratic readings of sensor and wrongly believes that the SGB-NP is working but it is not. The control command to stop SGB-NP and activate SGB-BP is not provided so SGB-NP is not stopped timely (SH.1, SH.2). It is assumed that this situation is recognized almost immediately, but the system must be shut down (L.1, L.2 and L.3) until it can be restored through maintenance.

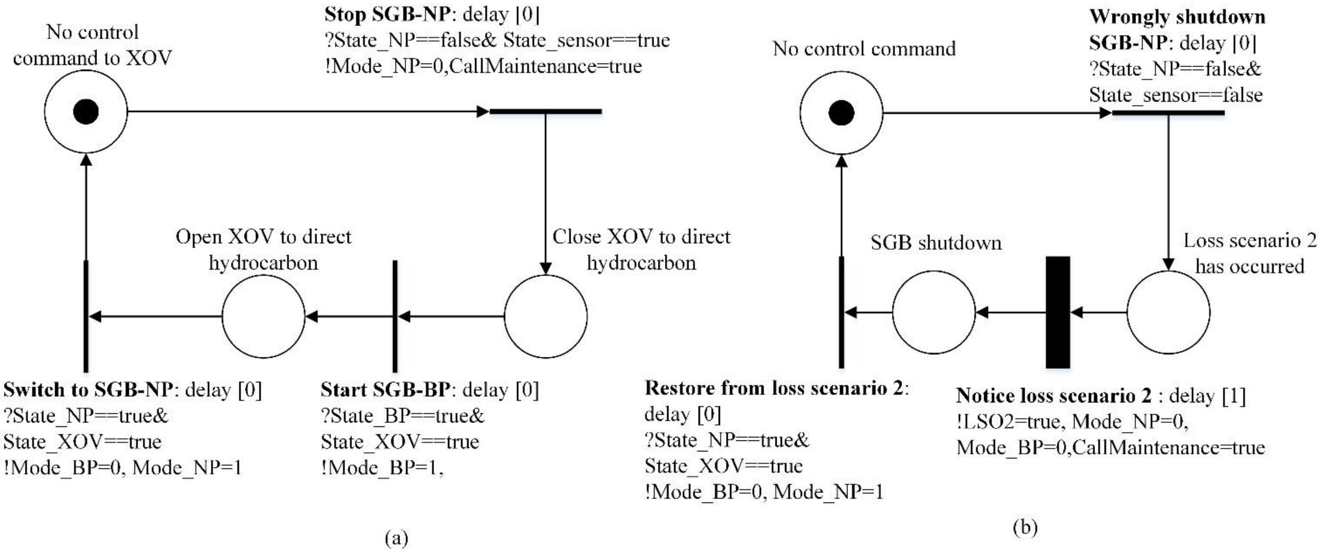

No control command to XOV
Stop SGB-NP: delay [0]
?State_NP==false& State_sensor==true
!Mode_NP=0,CallMaintenance=true
Open XOV to direct hydrocarbon
Close XOV to direct hydrocarbon
Switch to SGB-NP: delay [0]
?State_NP==true&
State_XOV==true
!Mode_BP=0, Mode_NP=1
Start SGB-BP: delay [0]
?State_BP==true&
State_XOV==true
!Mode_BP=1,
(a)
No control command
Wrongly shutdown SGB-NP: delay [0]
?State_NP==false&
State_sensor==false
SGB shutdown
Loss scenario 2 has occurred
Restore from loss scenario 2:
delay [0]
?State_NP==true&
State_XOV==true
!Mode_BP=0, Mode_NP=1
Notice loss scenario 2 : delay [1]
!LSO2=true, Mode_NP=0,
Mode_BP=0,CallMaintenance=true
(b)

Figure 7 Mapping safe scenario and loss scenario 2 into SPN models

Figure 7 illustrates SPN for the safe scenario in (a) and loss scenario 2 in (b). The safe scenario is that the control command is provided correctly to switch from SGB-NP to SGB-BP in presence of failure of SGB-NP. Once the failure has been detected, the preparation of maintenance can start (!*CallMaintenance*=true) and SGB-NP is stopped (!*Mode_NP*=0). If both SGB-BP and XOV are available, then the processing line is switched to SGB-BP (!*Mode_BP*=1). After maintenance is completed, hydrocarbon is redirected to normal processing line as the faulty SGB-NP, SGB-BP and XOV is replaced. The loss scenario 2 can occur when sensor provide incorrect feedback (?*State_sensor*==false) in together with failure on SGB-NP (?*State_NP*==false). This loss scenario is immediately detected after 1 hour and the system is shutdown (!*Mode_BP*=0, *Mode_NP*=0, *SO2*=true) and preparation of maintenance start (!*CallMaintenance*=true). After maintenance is completed, the system is restored in the same way as safe scenario. SPN model for loss scenario 1 can be also generated in the similar way. It is assumed that variables related to loss scenarios and safe scenario (State_sensor, State_XOV, State_NP, State_BP) are subjected to system evolution and interventions, which is described by the RAM model. The variables *Mode_BP* and *Mode_NP* indicate whether there are hydrocarbon flows into the available processing line or not. These two variables are defined in integral domain, whereas the other variables are defined in Boolean domain.

4.3 Step II: Develop RAM models for selected loss scenarios

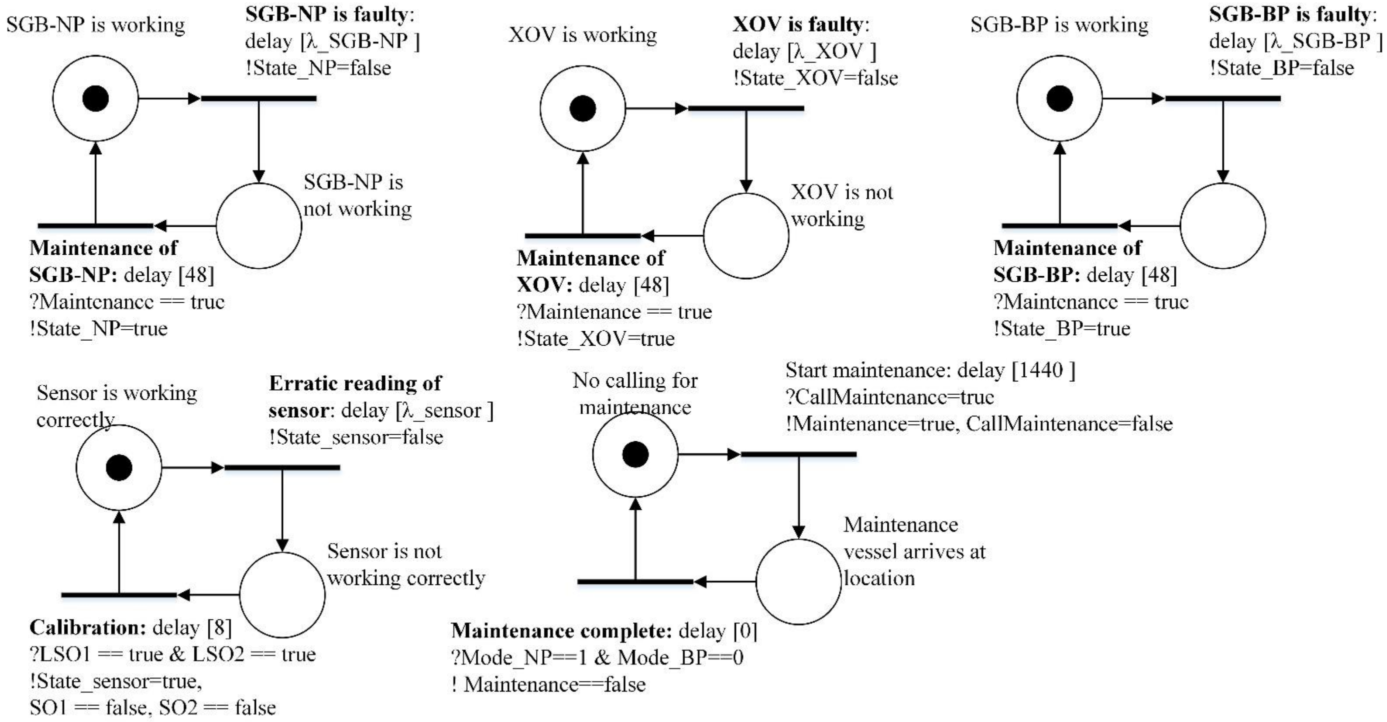

SGB-NP is working
SGB-NP is faulty: delay [λ_SGB-NP] !State_NP=false
SGB-NP is not working
Maintenance of SGB-NP: delay [48] ?Maintenance == true !State_NP=true
XOV is working
XOV is faulty: delay [λ_XOV] !State_XOV=false
XOV is not working
Maintenance of XOV: delay [48] ?Maintenance == true !State_XOV=true
SGB-BP is working
SGB-BP is faulty: delay [λ_SGB-BP] !State_BP=false
Maintenance of SGB-BP: delay [48] ?Maintenance == true !State_BP=true
Sensor is working correctly
Erratic reading of sensor: delay [λ_sensor] !State_sensor=false
Sensor is not working correctly
Calibration: delay [8] ?LSO1 == true & LSO2 == true !State_sensor=true, SO1 == false, SO2 == false
No calling for maintenance
Start maintenance: delay [1440] ?CallMaintenance=true !Maintenance=true, CallMaintenance=false
Maintenance vessel arrives at location
Maintenance complete: delay [0] ?Mode_NP==1 & Mode_BP==0 ! Maintenance==false

Figure 8 SPN model for maintenance and evolution of controlled process

Figure 8 presents SPN model for related variables. The maintenance of hardware component (i.e. SGB-NP, SGB-BP and XOV) is completed together after a certain delay (1440 hours), so the variable *Maintenance* is introduced to synchronize the maintenance events on different piecewise SPN. Since it is assumed that there is no means to reveal the erratic readings on sensor, the sensor is updated through on-line program after 8 hours once both loss scenarios have been recognized (?*LSO1*==true & *LSO2* ==true).

The reliability data for subsea equipment retrieved from the database OREDA [40] are re-evaluated based on discussion with system designer considering the novelty of technology and operating conditions. The estimated data, assumptions and computational setting are as follows:

1) The status of SGB-NP, SGB-BP and XOV is assumed to be under continuously monitoring, thus the failure is immediately revealed. The failure rates for SGB-NP, SGB-BP and XOV are assumed as 3×10^{-5} hour^{-1}, 1×10^{-5} hour^{-1} and 1.5×10^{-6} hour^{-1} respectively. All the failure events are assumed to be exponentially distributed. The sensor is assumed to continuously provide the feedback that is possibly erratic. To compare various control strategies, the four sets of transition rates for this failure mode are assumed as:
   - Case 0: occurrence rate for erratic reading =0
   - Case 1: occurrence rate for erratic reading =0.5×10^{-5} hour^{-1}
   - Case 2: occurrence rate for erratic reading =1×10^{-5} hour^{-1}
   - Case 3: occurrence rate for erratic reading =1.5×10^{-5} hour^{-1}
2) System run with 55% production efficiency when SGB-BP is active.
3) The time for mobilization is 1440 hours. The time of retrieval and reinstallation is delayed for 48 hours. The faulty equipment is replaced (as good as new after maintenance) and the working equipment keeps running as it is (as bad as old after maintenance).

4) The experiment time for simulation is 10 years (i.e. 87600 hours). 5×10^5 simulation runs have been used for each case. The computation time was approximately 44 minutes with a 2.60 GHz processor, 16 GB of RAM, and it can increase if there are more variables to observe.

4.4 Numerical results

The frequency of loss scenarios was calculated by observing the frequency of related transitions in SPN, as reported in Table 6. Loss scenario 1 only lead to SH.1, which in worst condition can lead to the production loss (L.1). Loss scenario 2 can lead to all three system-level hazards, which in worst condition can lead to production loss (L.1, L.3) and the hydrocarbon spills accident (L.2). The costs for associated consequence of L.2 given the emergency barrier management can be estimated through event tree analysis.

Table 6 Frequency of loss scenario 1 and 2

	Loss scenario 1 (L.1)	**Loss scenario 2 (L.1, L.2, L.3)**
Case 1	7.028×10^{-2} year^{-1}	3.3×10^{-4} year^{-1}
Case 2	1.427×10^{-1} year^{-1}	5.7×10^{-4} year^{-1}
Case 3	2.033×10^{-1} year^{-1}	7.9×10^{-4} year^{-1}

The effect of loss scenarios on production loss can be directly calculated through simulation. Figure 9 and Figure 10 illustrate the average value of system production deficiency and system unavailability from 0 to t, respectively.

The system production deficiency is stated as below:

$$100\% - (Mode\_BP \times 55\% + Mode\_NP)$$

And system unavailability equals to:

$$1 - (Mode\_BP + Mode\_NP)$$

Where the initial value for variable *Mode_NP* is 1, whereas *Mode_BP* is assumed to be 0 as bypass processing line is not working in the beginning of operation.

Case 0 shows the situation that the adequate control has been provided for loss scenario 1 and 2, therefore only the safe scenario has been considered. As reported in Table 6, the frequency of loss scenario 1 seems as proportional to the occurrence rate for erratic reading, whilst loss scenario 2 is not. The reason is that loss scenario 1 is subjected to unavailability of sensor (that is proportional to the occurrence rate for erratic reading) and availability of SGB_NP, whereas loss scenario 2 is subjected to unavailability of sensor and unavailability of SGB_NP. The availability of SGB_NP can be seen as proportional to the occurrence rate for erratic reading due to the impact of maintenance in both safe scenario and loss scenario 2, whilst unavailability of SGB_NP is not.

The average unavailability and production deficiency in case 0 are 0.0057 and 2.14%, whereas in worst case (case 3) are 0.0148 and 3.08%. If assume that SGB can produce 2 million kroner worth oil and gas per day or 730 million Norwegian kroner (NOK) per year, then the expected difference between case 0 and case 3 is 6.862 million NOK per year in stakeholder’s favor.

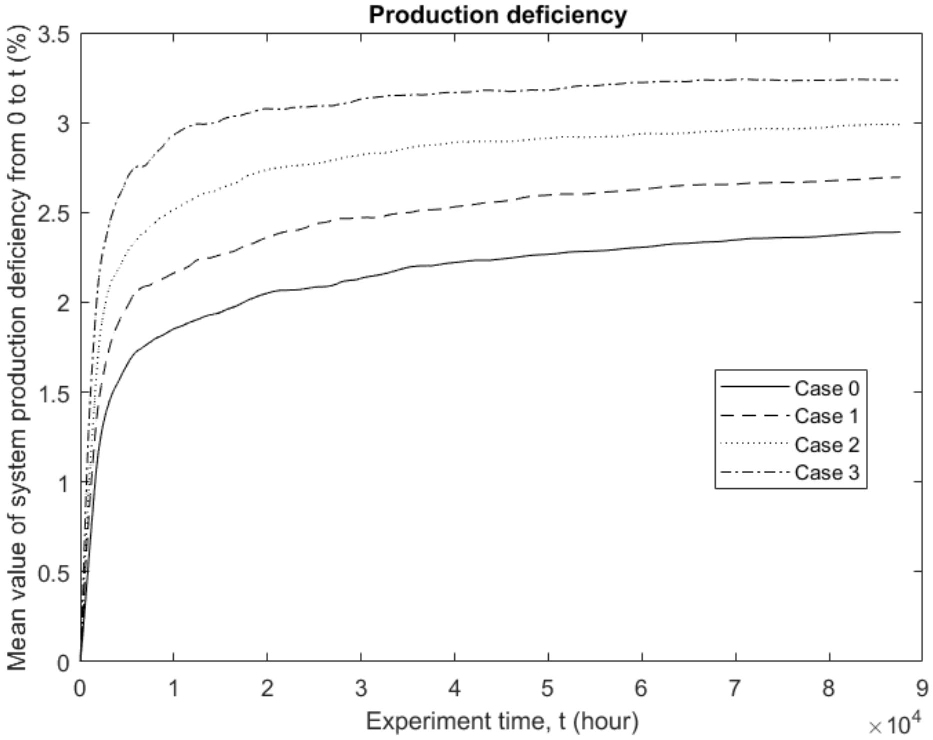


Figure 9 System production deficiency of case 0-3

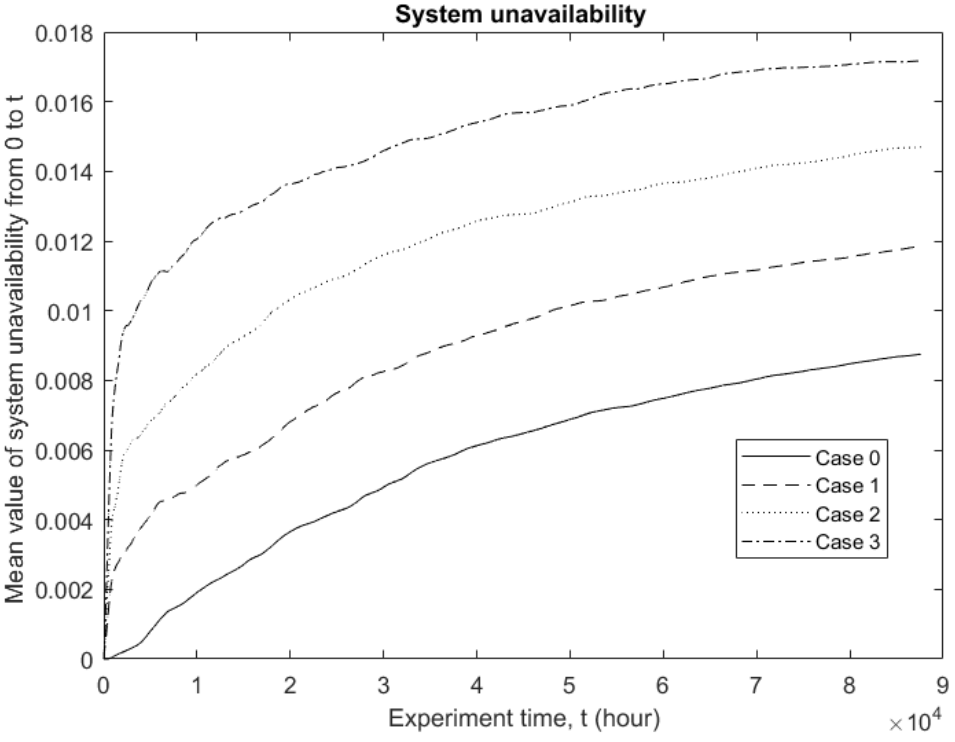


Figure 10 System unavailability of case 0-3

It is observed that the effect of loss scenarios is considerable, according to their impact on production and potential for severe accident like hydrocarbon spills. Some example countermeasures are suggested as following:

- Preventive countermeasure is to reduce the transition rate to the state that sensor has the erratic reading. For example, the validity and accuracy of signals from sensors can be increased by removing noise from piping conditions.
- Compensating countermeasure is to increase the ability of controller to discriminate between a real demand and false demand caused by erratic readings provided by sensor. For instance, installation of master sensor that monitors and compares the reading of duty sensor.
- One may also notice that the loss scenario 1 has much less severe consequence but high frequency than loss scenario 2. The system designer may consider to start troubleshooting once loss scenario 1 has been recognized. The premise condition for loss scenario 2 can be removed in this situation since they share the same casual factor and these two loss scenarios cannot occur simultaneously. This said, the hidden error in sensor is revealed and subsequently corrected by a demand.

The selection of compensating and preventive countermeasure depends on frequency of loss scenarios obtained through STPA-RAM modelling and the cost estimation for adverse effects and perceived benefits, where the later one is beyond the scope of this article.

5. DISCUSSION

This article proposes a new approach to combine STPA and RAM models, with support of existing modelling formalism SPN. The new approach is made of fundamental elements of each method, to take advantage of each strength whilst to compensate for their weakness. In this respect, the contributions are twofold: (1) to address uncertainty in STPA so its results can be confidently used by decision makers (2) to improve the construction of SPN model taking advantage of control structure offered by STPA.

5.1 Level of uncertainty

The proposed approach enables the quantification of hazards derived by a relatively new method STPA, and thereby improve the possibility for decision-making about design choices. It is reasonable to ask to what extent we have succeed in this respect. The level of uncertainty is of relevance for making such judgement. In this respect, uncertainty for STPA-RAM model can be categorized into *completeness uncertainty* that stems from stems from incomplete scope of hazard identification, *model uncertainty* that stems from low suitability of modelling formalism and *data uncertainty* that stems from improper selection of distribution and associated parameter values [41], as illustrated in Figure 11.

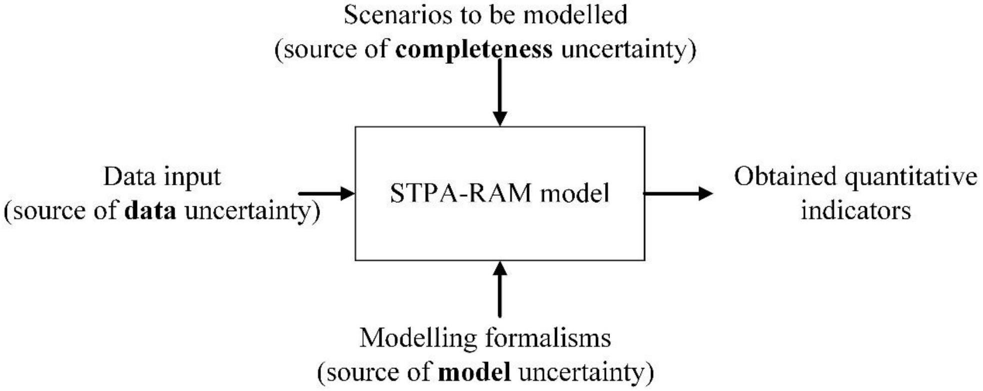

Scenarios to be modelled
(source of **completeness** uncertainty)
Data input
(source of **data** uncertainty)
STPA-RAM model
Obtained quantitative indicators
Modelling formalisms
(source of **model** uncertainty)

Figure 11 Uncertainty related to STPA-RAM model

As discussed earlier, human errors and software errors become visible in STPA when they are properly defined in the feedback control loop. This feature ensures STPA to develop a (theoretically) complete spectrum of scenarios, where the term complete of course depends on the purpose of analysis as done in step 1 of STPA. When the detailed study of STPA is conducted, it is often to get hundreds of UCAs and thousands of loss scenarios. It is practically impossible to include them in one single STPA-RAM model due to a significant increase in computational burden. The pre-processing methods for STPA-RAM model in this sense are required, for example to eliminate loss scenarios based on existing and planned safety barriers as suggested in [14], or to prioritize loss scenarios based on criticality or risk measures. If the rationales behind these pre-processing methods are specified and documented, the category of completeness uncertainty is reduced.

SPN with predicates and assertions can model loss scenarios without distorting the phenomenon of control structure. The reason is that the use of predicates and assertions using variables can introduce the guard for transitions, which is equivalently the context for safe or unsafe control actions. If the user of STPA-RAM model is competent and aware of the limitation of employing SPN, the model uncertainty of STPA-RAM model is well acknowledged.

The major bottleneck for STPA-RAM model seems to be data uncertainty. The reason is that the loss scenarios derived by STPA move beyond the failure scenarios as the combination of failure modes, whereas most of data resource collect and record data on basis of failure modes. The probabilistic modelling of loss scenarios is therefore greatly relied on the expert judgement and engineering experience. Rather than abandoning probabilistic model, Berner and Flage [42] elaborated a solution to evaluate the strength of background knowledge and beliefs about assumption deviations as supplement to the use of probability tools. The confidence or data uncertainty of STPA-RAM model therefore depends on the description of background knowledge that judges and justifies the judgement about assumptions and simplification made. This is remarked as the future work as the potential improvement to the proposed approach.

5.2 Pattern-wise model construction

When dealing with a complex system, it often occurs that a large scale SPN model is constructed and remains unreadable and unmanageable [36]. The reason may be the lack of proper description model before constructing SPN model so the construction mainly relies on the imagination of model designer. STPA in this sense can facilitate the model construction of SPN model. The behavior (e.g. failure) of components can be classically modelled by piecewise SPN model.

The remaining question is about how to model the complex maintenance process as control loops, especially for predictive maintenance with the enhanced level of digitalization. Here we propose to model such complex maintenance process as a feedback control loop advocated in STPA: the decision on maintenance is considered as a controller of some sort, the feedback for making decisions are for example the degradation level of component, the control action is therefore to change the state of components for example notifying personnel of maintenance/replacement of equipment. The complex maintenance process is then modelled as a pattern in SPN, for example as shown in Figure 8. The interfaces of maintenance process to other patterns are representing by global variables (e.g. Mode_BP and Mode_NP in Figure 8).

With such process, we can produce the patterns of different control loops and they can be replicated as many times as need, and make the large-scale SPN model more compact and understandable. By translating description model into SPN model, the causality knowledge can be traceable and updated when hierarchical control structure is updated (for example from step 2 to step 4 in an original STPA procedure). More importantly, when there is more than one hierarchical control structure, we can use the same process to synthesize them and complete in a one single model if necessary. In this regard, we argue that STPA can facilitate constructing SPN model, and this feature makes STPA-RAM model more appealing for systems with complex maintenance processes.

5.3 Limitation and constraints

One limitation of the case study is that the loss scenarios selected for the numerical experiment in this paper would normally be identified by traditional failure mode analysis methods. Several authors claim that STPA is able to identify more hazards than traditional failure modes identification method, with regard to software error and interaction type of hazards [8, 10, 18]. For example, one complex loss scenario for SGB design case could be: ‘human operator adjusts set point of choke valve too late during high pressure of hydrocarbon in the SGB bypass processing line, due to a long procedure taken before giving decision or SCU delays in the processing of command to adjust set point of choke valve’. This loss scenario can be prevented by either updating operating procedures (e.g. the procedure must be done within appropriate amount of time) or modifying the design (e.g. SCU must be able to process the control command immediately).

Another limitation of the case study is the intentional exclusion of software flaws and human errors. One reason behind is that human errors and software flaws are often judged as systematic factors that must be removed before operation as required in standards, e.g. functional safety standard IEC 61508 [43]. Another reason is the lack of relevant database, implies a great dependence on expert judgements and operational experience. If relevant data is available and the related loss scenarios are judged as critical (e.g. poor knowledge and operating experience), STPA-RAM model can include the effect of human and software error to evaluate how they contribute to the frequency of loss scenarios. The interesting part is that learning pattern for software and human [22]. It means that human or software can learn from experience, and same hazards are avoided under the associated context. Taking the Figure 4 (b) as example, the casual factors considered for loss scenario ‘sending control command too late’ could be the inadequate understanding of unscheduled situations occur. One can assume that the process model of controller can be improved through the lesson learnt. Therefore, the assertion of Tr5 is ‘!T=T×0.5’ to coarsely model this situation that the delay of detecting abnormal signal is decreased every time this loss scenario happens.

In case study, only two loss scenarios are modelled. Even some methods for elimination and prioritization of loss scenarios, the number of critical loss scenarios is likely to be more than that. Each loss scenario, or a combination of a few, is regarded as testing experiments of different operational situations. Despite our approach taken, it is interesting to investigate strategies for including more loss scenarios in the same model, when this is needed.

In some applications, the evolution of controlled process may be subjected to the shocks from environment, which is not modelled statistically. For instance, if the case study is further refined to study the performance on SPM, then the process variables like liquid level on separator is considered. This process variable is determined by the control command (e.g. open/close liquid discharge valve) and the environmental disturbance (e.g. flow conditions from wells). The change of state of latter one is less predictable than the first one that is subjected to stochastic event. The potential solution for this problem may be to integrate STPA-RAM model with the model that studies the physics of controlled process, e.g. finite element analysis. The simulation time is therefore greatly amplified by the agility of process variables, which make the proposed approach unappealing when comes to the industry-scale system.

6. CONCLUSION

This article has discussed the potential interface between STPA as qualitative analysis and RAM models as means for quantification. It is argued that qualitative analysis is still needed to interpret the loss scenarios derived from STPA. In this respect, an integrated approach that combines the STPA and RAM model through SPN that follow state-event transition formalism is proposed. In the case study, it has been shown that the STPA-RAM model can quantitatively calculate the frequency of loss scenarios by combing with prepared RAM models. The numerical results give risk-based insights to system production, maintenance and emergency management, and some countermeasures are suggested accordingly. We conclude that the proposed approach is a way to connect between STPA and RAM models. This approach helps to clarify to what extent STPA can contribute to decision-making in an engineering design, e.g. the design of safety barrier and IMR strategies.

Future work includes to evaluate the background knowledge and sensitivities of assumptions made for probabilistic models, so the confidence of STPA-RAM model can be judged by decision makers. Some approaches have been discussed in [42]. The next step is then to fuse it into the approach proposed in this article. In addition, current framework of proposed approach focuses primarily on the side of constructing model for simulation, but few attention has been paid to balance the simplicity and expressiveness of STPA-RAM model. Another important area of further research is therefore to develop approach to screen out and prioritize the loss scenarios. One possible strategy is to evaluate the effectiveness of safety constraints in terms of its availability and easiness of implementation, as well as the criticality of associated losses. This may require not only the advance in analytical methods, but also the multidisciplinary participations for conducting STPA to seek multiple perspectives for prioritization.

AKNOWLEDGEMENT

This work was carried out as a part of SUBPRO, a research-based innovation center within Subsea Production and Processing. The authors gratefully acknowledge the project support, which is financed by the Research Council of Norway, major industry partners and NTNU. Our special thanks to the industry partner DNV-GL inside SUBPRO, who contributes to this body of knowledge and provides inspirations on current issues facing in the post-processing of STPA.

REFERENCE

[1]. Bai, Y. and Q. Bai, *Subsea Engineering Handbook*. 2010, Boston: Gulf Professional Publishing. xxv.

[2]. Leveson, N., *Engineering a Safer World: Systems Thinking Applied to Safety*. 2011: MIT Press.

[3]. Abdulkhaleq, A., S. Wagner and N. Leveson, *A Comprehensive Safety Engineering Approach for Software-Intensive Systems Based on STPA.* Procedia Engineering, 2015. **128**: p. 2-11.

[4]. Rasmussen, J., *Risk management in a dynamic society: a modelling problem.* Safety Science, 1997. **27**(2): p. 183-213.

[5]. Seligmann, B.J., E. Németh, K.M. Hangos and I.T. Cameron, *A blended hazard identification methodology to support process diagnosis.* Journal of Loss Prevention in the Process Industries, 2012. **25**(4): p. 746-759.

[6]. Hollnagel, E., *FRAM:The functional resonance analysis method:modelling complex socio-technical systems*. 2012 Farnham: Ashgate Publishing Ltd.

[7]. Leveson, N. and J. Thomas, *STPA handbook* 2018: MIT.

[8]. Mahajan, H.S., T. Bradley and S. Pasricha, *Application of systems theoretic process analysis to a lane keeping assist system.* Reliability Engineering & System Safety, 2017. **167**: p. 177-183.

[9]. Kim, H., M.A. Lundteigen, A. Hafver, F. Pedersen and G. Skofteland, *Application of Systems-Theoretic Process Analysis to isolation of subsea wells: opportunities and challenges of applying STPA to subsea operation*, in *Offshore Technology Conference*. 2018: Houston, Texas, USA.

[10]. Sulaman, S.M., A. Beer, M. Felderer and M. Höst, *Comparison of the FMEA and STPA safety analysis methods–a case study.* Software Quality Journal, 2017.

[11]. Abdulkhaleq, A., D. Lammering, S. Wagner, J. Röder, N. Balbierer, L. Ramsauer, T. Raste and H. Boehmert, *A Systematic Approach Based on STPA for Developing a Dependable Architecture for Fully Automated Driving Vehicles.* Procedia Engineering, 2017. **179**(Supplement C): p. 41-51.

[12]. Faiella, G., A. Parand, B.D. Franklin, P. Chana, M. Cesarelli, N.A. Stanton and N. Sevdalis, *Expanding healthcare failure mode and effect analysis: A composite proactive risk analysis approach.* Reliability Engineering & System Safety, 2018. **169**(Supplement C): p. 117-126.

[13]. Nakao, H., M. Katahira, Y. Miyamoto and N. Leveson. *safety guide design of crew return vehicle in concept design phase using STAMP/STPA*. in *Proceeding of the 5th IAASS Conference* 2011. Citeseer.

[14]. Rokseth, B., I.B. Utne and J.E. Vinnem, *A systems approach to risk analysis of maritime operations.* Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 2016. **231**(1): p. 53-68.

[15]. Kim, H., M.A. Lundteigen, A. Hafver, F. Pedersen, G. Skofteland, C. Holden and S.J. Ohrem, *Application of Systems-Theoretic Process Analysis to a Subsea Gas Compression System*, in *ESREL* 2018: Trondheim, Norway.

[16]. Young, W. and N. Leveson, *Systems thinking for safety and security*, in *Proceedings of the 29th Annual Computer Security Applications Conference*. 2013, ACM: New Orleans, Louisiana, USA. p. 1-8.

[17]. Friedberg, I., 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, 2017. **34**: p. 183-196.

[18]. Rodríguez, M. and I. Díaz, *A systematic and integral hazards analysis technique applied to the process industry.* Journal of Loss Prevention in the Process Industries, 2016. **43**: p. 721-729.

[19]. Wróbel, K., J. Montewka and P. Kujala, *System-theoretic approach to safety of remotely-controlled merchant vessel.* Ocean Engineering, 2018. **152**: p. 334-345.

[20]. Hafver, A., S. Eldevik, I. Jakopanec, O.V. Drugan, F. Pedersen, R. Flage and T. Aven, *Risk-based versus control-based safety philosophy in the context of complex systems*. 2017. 38-38.

[21]. Rausand, M. and A. Høyland, *System Reliability Theory, Models, Statistical Methods, and Applications*. second edition ed. Hoboken, NJ. 2004: John Wiley & Sons, Inc. 419-464.

[22]. ISO/TR 12489, *Petroleum, petrochemical and natural gas industries -- Reliability modelling and calculation of safety systems*. 2013, Geneva: International Electrotechnical Commission.

[23]. Innal, F., *Contribution to modelling safety instrumented systems and to assessing their performance: Critical analysis of IEC 61508 standard, PhD Thesis*. 2008, Bordeaux: University of Bordeaux

[24]. IEC61165, *Application of Markov techniques*. 2006.

[25]. IEC 62551, *Analysis techniques for dependability - Petri net techniques*. 2012.

[26]. Marsan, M.A., G. Balbo, G. Chiola, G. Conte, S. Donatelli and G. Franceschinis, *An introduction to generalized stochastic Petri nets.* Microelectronics Reliability, 1991. **31**(4): p. 699-725.

[27]. Rauzy, A., *Guarded transition systems: A new states/events formalism for reliability studies.* Proceedings of the Institution of Mechanical Engineers. Part O, Journal of risk and reliability, 2008. **222**(4).

[28]. Signoret., J.-P., *Dependability & Safety Modeling and calculation: Petri Nets*, in *In Proceeding of the 2nd IFAC Workshop on Dependable Control of Descrete Systems*. 2009: Bari, Italy.

[29]. Wang, R., W. Zheng, C. Liang and T. Tang, *An integrated hazard identification method based on the hierarchical Colored Petri Net.* Safety Science, 2016. **88**: p. 166-179.

[30]. Dirk, S., H. René Sebastian, W. Jan and S. Eckehard, *Integration of Petri Nets into STAMP / CAST on the example of Wenzhou 7.23 accident.* IFAC Proceedings Volumes, 2013. **46**(25): p. 65-70.

[31]. Rokseth, B., I.B. Utne and J.E. Vinnem, *Deriving verification objectives and scenarios for maritime systems using the systems-theoretic process analysis.* Reliability Engineering & System Safety, 2018. **169**(Supplement C): p. 18-31.

[32]. Leveson, N. and J. Thomas, *STPA primer version 1*. 2013.

[33]. Bjerga, T., T. Aven and E. Zio, *Uncertainty treatment in risk analysis of complex systems: The cases of STAMP and FRAM.* Reliability Engineering & System Safety, 2016. **156**(Supplement C): p. 203-209.

[34]. Thomas, J., *Extending and Automating STPA for Requirements Generation and Analysis, Ph.D. Dissertation*. 2013: MIT.

[35]. Balbo, G., *Introduction to Generalized Stochastic Petri Nets*, in *Formal Methods for Performance Evaluation: SFM*, M. Bernardo and J. Hillston, Editors. 2007, Springer Berlin, Heidelberg. p. 83-131.

[36]. Signoret, J.-P., Y. Dutuit, P.-J. Cacheux, C. Folleau, S. Collas and P. Thomas, *Make your Petri nets understandable: Reliability block diagrams driven Petri nets.* Reliability Engineering & System Safety, 2013. **113**: p. 61-75.

[37]. SUBPRO, *Subsea production and processing*. 2015: https://www.ntnu.edu/subpro.

[38]. Diaz, M.J.C., M. Stanko and S. Sangesland, *Exploring New Concepts in Subsea Field Architecture*, in *Offshore Technology Conference*. 2018.

[39]. GRIF, *Graphical Interface for reliability Forecasting*. 2016, France: SATODEV.

[40]. OREDA, *Offshore and Onshore Reliability Data, 6th edition*. 2015.

[41]. NUREG-1855, *Guidance on the treatment of uncertainties associated with PRAs in risk-informed decision making*. 2009: Nuclear Regulatory Commision

[42]. Berner, C. and R. Flage, *Strengthening quantitative risk assessments by systematic treatment of uncertain assumptions.* Reliability Engineering & System Safety, 2016. **151**: p. 46-59.

[43]. IEC 61508, *Functional safety of electrical/electronic/programmable electronic safety related systems. Part 1-7*. 2010, Geneva: International Electrotechnical Commission.