Scientific Research Publishing

Journal of Software Engineering and Applications, 2017, 10, 211-244
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Developing Dependability Requirements Engineering for Secure and Safe Information Systems with Knowledge Acquisition for Automated Specification

Mohammed Abu Lamddi

Software Engineering Department, University of Palestine, Gaza, Palestine
Email: m.abulamddi@up.edu.ps

How to cite this paper: Lamddi, M.A. (2017) Developing Dependability Requirements Engineering for Secure and Safe Information Systems with Knowledge Acquisition for Automated Specification. *Journal of Software Engineering and Applications*, **10**, 211-244.
https://doi.org/10.4236/jsea.2017.102013

Received: November 29, 2016
Accepted: February 25, 2017
Published: February 28, 2017

Abstract

Our dependability on software in every aspect of our lives has exceeded the level that was expected in the past. We have now reached a point where we are currently stuck with technology, and it made life much easier than before. The rapid increase of technology adoption in the different aspects of life has made technology affordable and has led to an even stronger adoption in the society. As technology advances, almost every kind of technology is now connected to the network like infrastructure, automobiles, airplanes, chemical factories, power stations, and many other systems that are business and mission critical. Because of our high dependency on technology in most, if not all, aspects of life, a system failure is considered to be very critical and might result in harming the surrounding environment or put human life at risk. We apply our conceptual framework to integration between security and safety by creating a S*a*S (Safety and Security) domain model. Furthermore, it demonstrates that it is possible to use goal-oriented KAOS (Knowledge Acquisition in automated Specification) language in threat and hazard analysis to cover both safety and security domains making their outputs, or artifacts, well-structured and comprehensive, which results in dependability due to the comprehensiveness of the analysis. The conceptual framework can thereby act as an interface for active interactions in risk and hazard management in terms of universal coverage, finding solutions for differences and contradictions which can be overcome by integrating the safety and security domains and using a unified system analysis technique (KAOS) that will result in analysis centrality. For validation we chose the Systems-Theoretic Accident Model and Processes (STAMP) approach and its modelling language, namely System-Theoretic Process Analysis for safety (STPA), on the safety side and System-Theoretic Process Analysis for Security (STPA-sec) on the security side in

order to be the base of the experiment in comparison to what was done in S*a*S. The concepts of S*a*S domain model were applied on STAMP approach using the same example *@RemoteSurgery*.

Keywords

Safety Information Model, Security Information Model, Dependability Requirements, Goal Modeling, KAOS, Obstacles Base, Risk Management

1. Introduction

The high level of integration between safety and security has widespread recognition that can be benefited from, in spite of the differences and similarities between the two, in other fields although members of these fields do not interact enough neither with members from the same field or members from other fields. This insufficient interaction is associated with requirements upon the associated architectural mechanisms. Security engineering is the practice of mechanisms, measures, and counter policies against potential risks such as encryption, firewalls, and backup, unlike safety engineering which is different in definition, practices [1].

Challenges such as concept, tools, and methods used in the fields of safety and security, arise during research on either field. The big gap between the two fields is resulted from the fact that research focused on either one of these two fields given that each of which has its own development tools and methods. To clarify this, the following example from the literature can be used; in a building, the safety engineer sees the emergency door important and should be accessible in cases of emergency. On the other hand, security engineer sees the emergency door a loophole that can provide access to the building to unauthorized personnel and therefore must be secured. However, the requirements of safety and security are similar in the fact that they are concerned about what the system-to-be should and should not do.

Although this is the typical way to distinguish the two fields, it also exist another distinction: "*Security is concerned with the risks originating from the environment and potentially impacting the system*", whereas "*safety deals with the hazard arising from the system and potentially impacting the environment*" [2]. Common for both fields is the risk see (**Figure 1**), which expresses the potential of harm, mostly stated through probability and severity. It is important, both during the system development and operations, to identify, analyse, evaluate and finally deal with as many relevant risks as possible. At the same time there are different techniques used within the fields, especially as safety deals with *unintentional hazards* and security with *intentional threats*.

Safety and security model has been focused on from different perspective and areas. Some researchers focused on the architectural framework while others focused on narrowing down the gap between the definitions and terminology

Figure 1. The red area represents our scope in the article.

adaptation in both safety and security or narrowing down between techniques and tools used in the system development life cycle.

In [3] researchers focused on security side only in a try to build a conceptual framework that deals with definitions and terminology related to security requirements from the beginning of the system development life cycle especially during the elicitation phase and requirement analysis which is reflected on the conceptual framework the researches put as a base for comparison and focused on analysing the methods used like Common Criteria, Secure Tropos, ISSRM, SREP, MSRA, Problem Frames, and the methods that depend on UML to the extent that these methods coverage in relation of the system development life cycle when used.

Avizienis *et al.* [4] addressed taxonomy of dependable and security by defining dependability from the security perspective and explained the means that could help achieve dependability in security. Furthermore, the researchers focused on taxonomy of threats, taxonomy of faults, and pathology of failure in the sense of explaining the terminologies but did not reflect them on a model.

Firesmith [5] [6] addressed the terminology of the taxonomy of safety and security as addressed by other researchers but what makes his researches different is that he focused on narrowing down the gap between safety engineering and security engineering through the implementation of an information model that relies on integrating and linking between safety and security while maintaining survivability and established underlying foundational concepts between the them and the concepts and relations using UML. Furthermore, in his latest work he redefined safety engineering and security engineering and from his definitions, the size of the comparison is clearly shown in the definitions he proposed and has also worked on enhancing it in tutorials [7].

Mayer *et al.* [8] propose a security requirements engineering process that consists of the following four steps: *Context analysis and asset identification*, *security goal determination*, *refinement of these goals to security requirements*, *and countermeasures selection*. Both of the latter two steps are based on a risk analysis approach named model-based information system security risk management (ISSRM). Thereby, Mayer *et al.* [8] propose to make use of Yu's i* [9] [10] requirements engineering techniques, which can also be used to deal with security requirements [11]. The proposed method by Mayer *et al.* [8] comprises security

requirements elicitation driven by a risk analysis method. It also supports analyzing security requirements through context and asset analysis.

Pièrre and Bouissou [12] focused on finding new methods to deal with modeling safety and security interdependencies with BDMP, a technique that depends on graphical modeling and mathematical formalism. However, using this newly founded method is impractical because it requires knowledge and hands-on experience because it is very much similar to attach tree and fault-tree. The newly founded technique was derived from a real case study used in [13] where the focus was on modeling the Stuxnet Attack with BDMP hoping towards more formal risk assessments.

Summary of the Contribution in Four Steps

Step 1: We propose a solution through the creation of a (S*a*S) Safety and Security information domain model that integrates safety and security domains, giving a better opportunity for comparison and integration to find a middle ground between the two domains, as well as unifying definitions through their mappings onto the common concepts.

Step 2: KAOS modelling language were used in running the example *@RunningSurgery* on the security and the safety sides in respect to both the S*a*S domain model and the hazard management process and we did the alignment of the S*a*S information domain model elements and the KAOS modelling language.

Step 3: We chose the Systems-Theoretic Accident Model and Processes (STAMP) approach and its modelling language, namely System-Theoretic Process Analysis for safety (STPA), on the safety side and System-Theoretic Process Analysis for Security (STPA-*sec*) on the security side in order to be the base of the experiment in comparison to what was done in steps 1 and 2.

The concepts of S*a*S domain model were applied on STAMP approach using the same example *@RemoteSurgery*.

STPA modelling language were used in running the example *@RunningSurgery* on the safety side in respect to both the STAMP domain model and the STPA hazard management process.

STPA-*sec* modelling language were used in running the example *@RunningSurgery* on the security side in respect to both the STAMP domain model and the STPA-*sec* hazard management process.

Step 4: We now have the S*a*S domain model and its own modelling language, KAOS-S*a*S, which resulted from the steps 1 and 2. We also have STAMP approach and its modelling language, STPA and STPA-*sec* that resulted from step 3.

Each domain and its own modelling language has been explained along with usage and execution on the same example *@RemoteSurgery* followed by the comparison and validation on how and to what extent each domain and its modelling language are covering the safety and the security sides.

2. Our Approach

Risk management process was the entry point for the integration process (**Figure**

1) as the interface interplays between safety requirements using information safety risk management domain model (ISRM) and security requirements using information system security risk management domain model (ISSRM) from the aspect of system functionality and what the system should and should not do. For that, we proposed the creation of an information domain model that integrates between safety and security, (S*a*S), and the implementation of risk management process that leads to dependability requirements for safety and security.

We will address the domain model of information safety risk management ISRM [5] [14] model by adding definitions for each artifact and adjusting it to comply with the work being done. Also we will address the Information System Security Risk Management ISSRM model [15].

This article presents an approach for applies a structured method to integration between security and safety by creating a (S*a*S) Safety and Security domain model. Furthermore, it demonstrates that it is possible to use goal-oriented (KAOS) Knowledge Acquisition in automated Specification language in threat and hazard analysis to cover both safety and security domains making their outputs, or artifacts, well-structured and comprehensive, which results in dependability due to the comprehensiveness of the analysis. The structured approach can thereby act as an interface for active interactions in risk and hazard management in terms of universal coverage, finding solutions for differences and contradictions which can be overcome by integrating the safety and security domains and using a unified system analysis technique KAOS that will result in analysis centrality.

We apply alignment between the safety and security (S*a*S) domain models for risk management with the modeling language KAOS, which has given the possibility for a better method to derive safety and security requirements in early stages from the beginning of the system development life cycle. The alignment between S*a*S domain model and KAOS enhances the cooperation and facilitates communication and interaction between stakeholders.

This article is composed of seven sections, two appendices and external file as supplementary material.

Section 3, titled "*Safety Engineering*" addresses the standards followed by our contribution in adapting information safety risk management (ISRM) domain to support the hazard management process, KAOS-Safety modelling languages and as a part of contribution we did the alignment between ISRM and KAOS.

Section 4, titled "*Security Engineering*" addresses the information system security risk management (ISSRM) domain, and followed by the risk management process, KAOS-Security modelling languages and alignment between KAOS and ISSRM domain.

Section 5, titled "*Safety and Security Engineering* (*SaS*)", is the result of the main contribution in integrating Section 3 and 4. We addressed the S*a*S domain produced followed by hazard/risk management process, KAOS-S*a*S modelling languages and running example *@RemoteSurgery* (**Appendix 1**). The example is run on S*a*S domain, the alignment between KAOS and S*a*S. The results of this

process are discussed in section 6.

Section 6, titled "*Validation*", consists of the validation and comparison between the uses of KAOS in running example on the suggested S*a*S domain and the use of Systems-Theoretic Accident Model and Processes (STAMP; **Appendix 2**) techniques languages (STPA; *STPA-sec*; supplementary material section) running the same example (section 5) on S*a*S domain.

Section 7, we provide our conclusions of the study and further work.

Appendix 1, this appendix describes the running example *@RemoteSurgery*.

Appendix 2, this appendix addresses the alignment between the concepts of STAMP Approach and S*a*S domain model and detailed explanation on the use of STAMP approach concept using the S*a*S domain model in running the example *@RemoteSurgery*.

Supplementary material (Technical Report) contains section titled "*STPA process for safety*" running example *@RemoteSurgery* System-Theoretic Process Analysis for safety (STPA) safety corner. This appendix is an extension to **Appendix 2**, the STAMP approach, where we use STPA Process for safety in running the example *@RemoteSurgery* that has been discussed in section 6 by KAOS, in the safety side section. We also use the same description of running example *@RemoteSurgery* and run it on the safety side using STPA Safety. The results of this process are discussed in section 6.

Also the external report contains section titled "*STPA-sec process for security*" running example *@RemoteSurgery* System-Theoretic Process Analysis for security (STPA-sec) security corner. This appendix is an extension to **Appendix 2**, the STAMP approach, where we use *STPA-sec* Process for security in running the example *@RemoteSurgery* that has been discussed in section 6 by KAOS, in the security side section. We also use the same description of running example *@RemoteSurgery* and run it on the security side using *STPA-sec*. The results of this process are discussed in section 6.

3. Safety Engineering

We will addresses the domain model of information safety risk management (ISRM) followed by our contribution in adapting information safety risk management (ISRM) domain to support the hazard management process, KAOS-*Safety* extension and as a part of contribution we did the alignment between ISRM and KAOS.

3.1. Domain Model of Information Safety Risk Management (ISRM)

Firesmith [5] distinguishes particularly harm coming from *Intentional* and *Unintentional* source. He then introduces the artifact of *defensibility* that is defined as the composition of both safety and security, and that is therefore closely related to the scope of our work.

The researchers Axelrod and Mayer commented on Information Safety Risk Management domain model ISRM [5]. Mayer [15] said that the ISRM domain does not deal with risk management process while Warren [16] argued that the

concepts of this domain, especially the description of the definitions *intentional* and *unintentional* and said that the safety domain should prevent the harmful impact of both accidental and intended hazardous events rather than protect individuals from harm.

The reason behind building ISRM domain model is trying to narrow between it and already existing models of security, which will be demonstrated in the security engineering section. The safety domain model is easily amenable to hazard analysis and supporting requirement engineering. (**Figure 2**) shows basic definitions on safety engineering like risk, hazard, accident, asset, and vulnerability that have a strong bond with requirement engineering definitions like safety goal, policy and requirement. This explains the public safety and risk analysis methodologies in terms of vulnerabilities, hazards, accidents, and assets.

Details on definitions the concepts of domain model ISRM (**Figure 2**) can be found in [5] [7][1].

3.2. ISRM Hazard Risk Management Process

Information safety domain model put by Firesmith [5] that addresses safety engineering and the creation of a *conceptualised* domain model specific for safety and discussed its concepts. He had also done the same for security integrated them into what he called survivability engineering. These domains are built similarly to the system development life cycle as it mainly depends on regular activities of requirement engineering for both safety engineering and security engineering. However, the steps or the risk management processes produced by Firesmith are not clear in the information models.

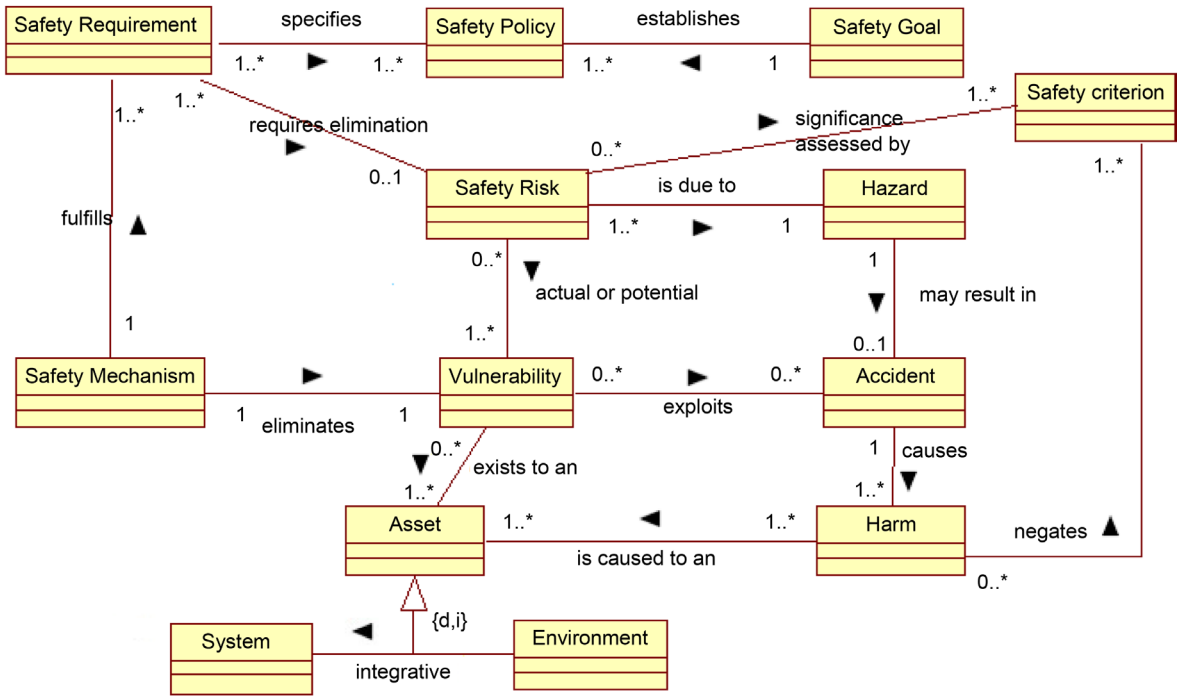

Figure 2. ISRM domain model adapted from [5] [14].

[1]We do not intend to provide absolute definitions of terms safety.

We elicited these six steps process (**Figure 3**) for risk management from the safety perspective through [16], standard IEC 61508 [17] and IEC 1508 standard [18] that explain the phases of the hazard risk management process from the safety perspective taking into account the respect to the safety information models [5]. The following steps are (1 to 6) summarised are follows:

1) Scope and asset identification the first step consists of the process of searching for stakeholders to address the safety implications, at the system level and their environments (a.k.a. *physical*, *social*, *standards*) for the purpose of defining the scope. After that, the assets of value for the company as well as the assets related to safety engineering need to be identified. The output of this step is the definition of the scope and its relation to the system and the environment and a priority list and rankings of assets to be secured from a safety perspective starting with the assets of the highest priority.

2) Determination of quality factor objective in this step, we set a quality criterion for every asset identified in the previous step, while each asset has its own characteristics, which requires the identification of safety goals for each of these assets.

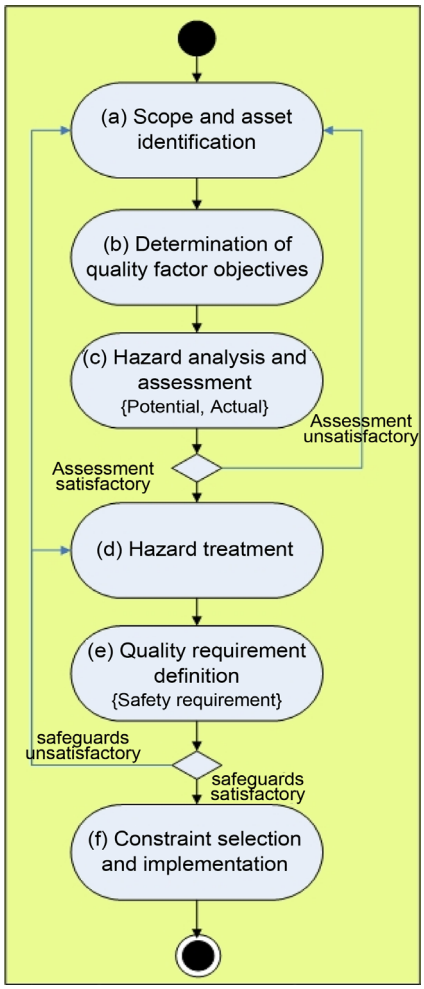

Figure 3. Hazard risk management process adapted from [15] [16].

3) Hazard analysis and assessment the third step consists of the identification of existing and potential hazards that are likely to violate the safety goals resulting in accidents. Without doubt, these accidents will cause damage to assets. After identification, these hazards are evaluated and the degree of risk is measured using quantitative and qualitative analysis. At this stage, the defining the likelihood of occurrence, defining consequence categories, and risk matrix are produced and the result is full information on these hazards. After that, ALARP principle is implemented to measure the tolerance of each hazard [17]. If the results are dissatisfying, the entire process has to be performed again starting from 1, otherwise, the process proceeds to 4.

4) Hazard treatment in this step the decision is made regarding these hazards. These types of risk treatments are divided to three categories: prevention, reducing, or retaining risk.

5) Quality requirements definition depending on the decision(s) made and choosing the measures in the previous step, we derive the safety mechanism, and the strategic decision that will satisfy safety requirement to define Safety Integrity Level (*SIL*) target that complies with what has been chosen in order to mitigate and control harm resulting from hazards.

6) Constraint selection and implementation in this step, the decisions made regarding hazards are implemented by setting constraints that comply to SIL target in parallel with implementing safeguards for unintentional hazards. To ensure the compatibility of the chosen quality criterion for each asset individually by referring to the safety policy.

Safety systems are dynamic and interactive resulting in having unintentional hazards. The upgrading process is continuous as the main objective of this step is to monitor the residual risk and its compliance to the standards [18].

3.3. Safety Modelling Languages

This section addresses KAOS for safety as well as artifact *Safety Obstacles* (Hazard).

In software engineering, requirements specifications are documents that describe what a system has to perform in order for the stakeholders needs from a new software system to be met.

3.3.1. KAOS for Safety

For safety requirements, it is very important to deal with *Obstacles* (Hazard) KAOS element, which capture undesired properties. It allows analysts to identify and address exceptional circumstances during requirements engineering in order to produce robust or new requirements to avoid or reduce the impact of *obstacles* giving more reliable software [19].

The more specific the goal is the more specific its obstructing obstacles will be. As mentioned earlier, a high-level goal produces high-level obstacles that will be refined into much smaller sub-obstacles. These *sub-obstacles* are used for precise obstacle identification in order to evaluate their feasibility through agent behaviour negative scenarios. It is much easier and preferable to refine what is

wanted than what is not wanted.

The level of how extensive obstacle identification is depends on the type and priority of the obstructed goal. For example, obstacle identification in *Safety Goals* needs to be adequately extensive. Domain-specific cost-benefit analysis needs to be performed to decide when the obstacle identification process should terminate.

Obstacle OR-refinement yields sufficient sub-obstacles to establish the obstacle; each OR-refinement of an obstacle obstructs the goal obstructed by this obstacle, goals and *AND*/*OR* refinement of obstacles proceed exactly the same way except for only a few alternative OR refinements are generally considered, in the case of obstacles, one may identify as many alternative obstacles as possible.

3.3.2. Alignment between KAOS Safety and ISRM Domain Model

In this section we will contribute towards Alignment between KAOS and ISRM and create relationship and mapping between the concepts of both KAOS and ISRM. Discussion about the name of the concepts is included in the ISRM domain model.

After identifying the different terms used in each ISRM source, our assumption that the terminology in the ISRM model is not unified has been validated. Many different terms are used to depict the same concept. More than a dozen of different names have been found for some concepts in Table 1 (concept (5) and (9)). Sometimes, the same name is used to depict different concepts. For example, Harm is due to an *accident* when dealing with *safety* engineering, is due to an attack when dealing with *security* engineering.

The process of extraction and concepts identification (Table 1) based on ISRM domain model and the definitions used for each concept in the ISRM model.

Table 1. Names of the concepts included in the ISRM model.

Type	Concept	Name from [5] [20]
Asset-related concepts	1	Asset Systems Environment
	2	Quality sub-factor; Safety criteria
Risk-related concepts	3	Risk; Safety risk;
	4	Harm
	5	Danger; Hazard; Accident
	6	Safety vulnerability
Risk treatment-related concepts	7	Safety Goal
	8	Safety requirement
	9	Safety mechanism; Safeguard; Safety tactic

After identifying the concepts comes the aligning process (Table 2), and define relationships between concepts of each model.

Asset-related concepts, KAOS is mainly focused on the security of the system-to-be, but it does not make a separation between the IS and business aspects. Thus, we align the Asset ISRM concepts concerning assets with the KAOS strategic goal, requirement and expectation (Table 2). Moreover, their *operationalisation* in operation and object are also assets. In KAOS, states of the system-to-be are described using object attributes. The purpose of the Safety goals is to achieving a target level of safety or one of its sub-factors (Table 3). In terms of KAOS, this means that the safety goals should define quality sub-factor (Table 3), and object attributes, which are concerned by potential risk events and hazard [21]. Thus, we align both safety goals and object attributes concerned by goal with ISRM quality sub-factor.

Risk-related concepts, In (Table 2), we align together ISRM *danger*, *hazard and threat* with KAOS *Negative scenarios*, *anti-goal* (also called malicious *obstacle*). *Anti-goals* can be identified at various abstraction levels, so they might need to be refined until they become *anti-requirements* or *anti-expectations* (assigned to an *anti-agent*). At higher abstraction levels, an *anti-goal* might be considered as the event, which, according to the ISRM model, is a combination of a hazard and one or more vulnerabilities (*safety*). At lower abstraction levels, an *anti-goal* (*anti-requirement or anti-expectation*) is a *hazard*, which is a potential attack or incident to assets. The language concepts for *anti-goal*, *anti-requirement* and *anti-expectation* remain respectively *goal*, *requirement and expectation*.

In (Table 2), we align ISRM safety vulnerability and the KAOS domain property.

Table 2. Concept alignment between KAOS extended to safety and the ISRM model.

ISRM model [5]		KAOS extended to safety	
		Synonyms in [21]	Language concept (modeling construct)
Asset-related concepts	Systems; Environment	Asset	Strategic Goals, Requirement, Expectation, Operation, Object
	quality sub-factor; Safety criteria	Safety Goal	Goal, Object attribute
Risk-related concepts	Risk; Safety risk;	/	/
	Harm	/	/
	Danger; Hazard; Accident	Hazard Obstacle;	Goal, Requirement, Expectation (in anti-model)
	Safety vulnerability	Vulnerability, domain property	Domain property
Risk treatment-related concepts	Safety Goal	Countermeasures	/
	Safety requirement	Safety-goal; Safety requirement; Safety expectation	Goal, Requirement, Expectation
	Safety mechanism; Safeguard; Safety tactic	/	New model implementing security components.

Table 3. Dependability attributes of S*a*S adapted from [20] [24].

Concepts Criterion S*a*S	
Safety	Security
Fail-safe	Confidentiality
Failure tolerance	Integrity
Performance	Availability
Robustness	
Correctness	
Accuracy	
Traceability	
Recoverability	
Human backup	

The KAOS domain property is a hypothesis about the domain that holds independently of the system-to-be. In correspondence, ISRM vulnerability (*Safety*) is defined as attributes of assets. Following the ISRM model, Hazard (Danger) cause harm to the assets, due to an accident when dealing with safety engineering, is due to an attack when dealing with security engineering.

ISRM domain model does not address hazard agent or hazard method. This explains why in this model, there is no description for agent and *operationalisation*.

Risk treatment-related concepts, ISRM risk treatment corresponds to the countermeasures [19] [21] that are elaborated after identification of the anti-goals. Countermeasures are not KAOS modeling concepts, but rather modeling idioms or patterns adopted by modelers. In KAOS, the countermeasures usually result in new safety goals, which need to be refined further into realisable safety requirements and expectations.

In (**Table 2**), we align ISRM safety requirement and the KAOS Safety goal (requirements and expectations). The refinement and *operationalisation* of the new safety goals, their concerned objects and attributes, and their assignment to agents (a.k.a *software*; *people*; *sub-system*), lead to new system-to-be components realising the necessary safety means. With respect to the ISRM model, these new system components correspond to *Safety mechanism*, *Safeguard* and *Safety tactic* [5].

4. Security Engineering

We will address the information system security risk management domain (ISSRM), and followed by the risk management process, KAOS-*Security* extension and alignment between KAOS and ISSRM domain.

4.1. Domain Model of Information System Security Risk Management (ISSRM)

Information System Security Risk Management ISSRM is a methodology that focuses on issues related to information systems security risk management. The model is defined after surveying risk management, the security related stan-

dards, risk management methods, and software engineering [15] [22]. The domain model shown in (**Figure 4**) supports security modelling languages alignment that also improves security and modelling languages because it is compatible with security threat management for organisations.

Details on definitions the concepts of domain model ISSRM (**Figure 5**) can be found in [15][2] on three levels, *Asset-related concepts*; *Risk-related concepts*; *Risk treatment-related concepts* in details.

4.2. ISSRM Risks Management Process

The ISSRM domain model is responsible for the risk assessment management process through three main concepts discussed each separately by Mayer [15] and they are as follows: Asset-related concepts; risk-related concepts; and risk treatment concepts. Using these three concepts, Mayer has put six steps (**Figure 6**) for the risk management process for the security requirement engineering. The following steps are (1 to 6) summarised as follows.

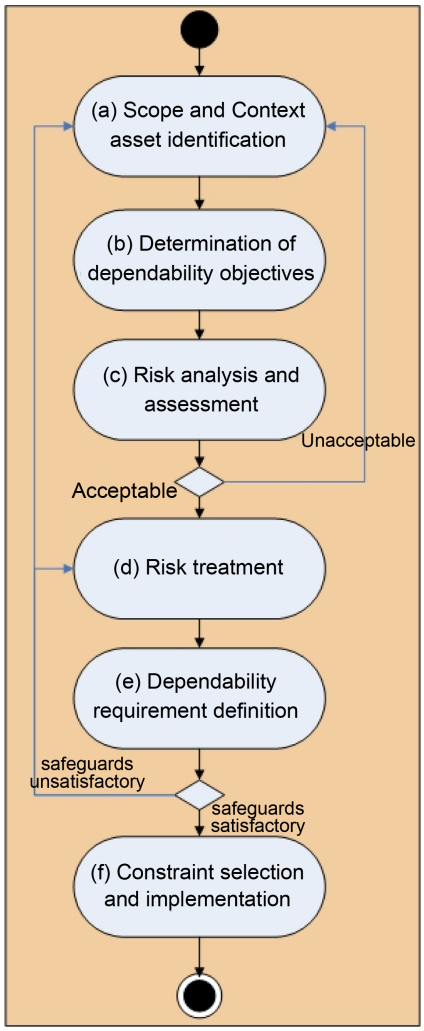

Figure 4. S*a*S risk management process, adapted from [25] [26].

[2]We do not intend to provide absolute definitions of terms security.

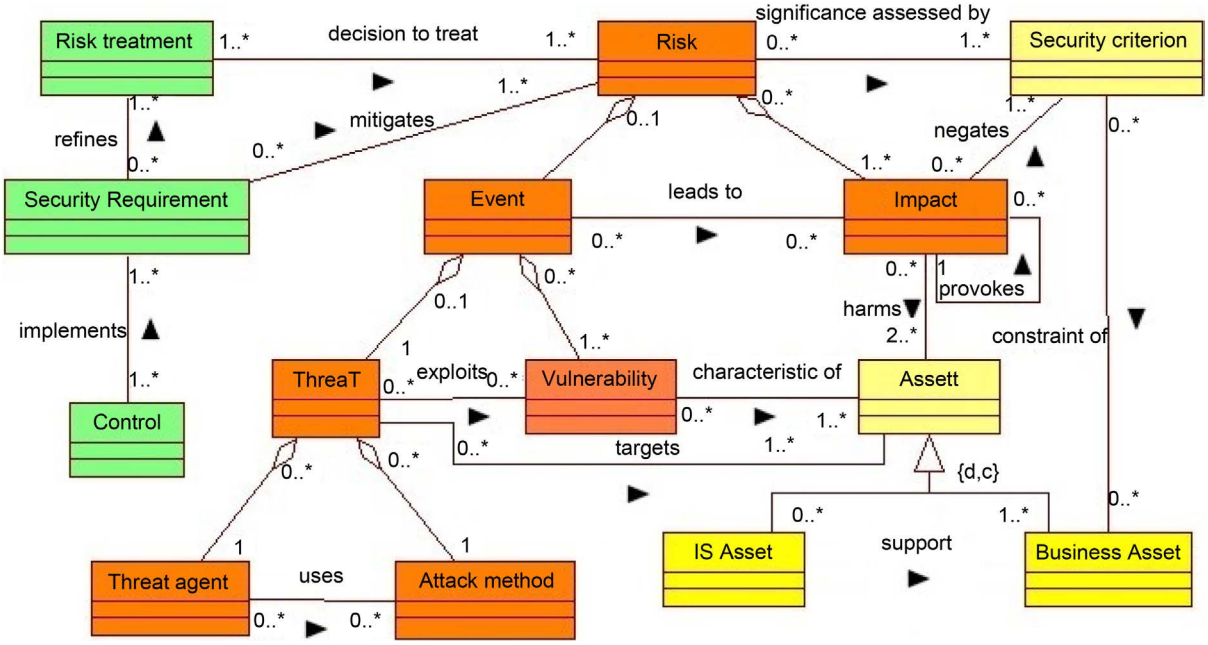

Figure 5. ISSRM domain model adapted from [15].

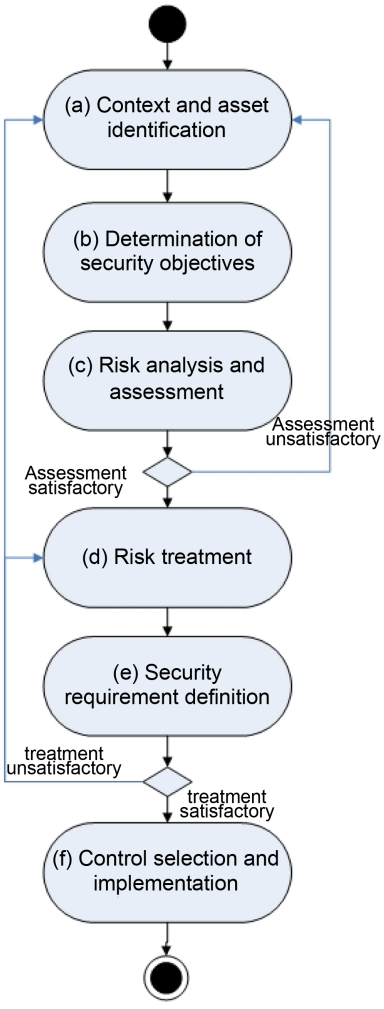

Figure 6. ISSRM process adapted from [15].

1) Context and asset identification the first process in this step is the search by multiple specialised teams for what is considered valuable for the company such as business assets and IS assets and what the processes the company wants to protect are. Ideally, a priority list of the assets that need security protection where said assets are arranged from the most important and are assigned the highest priority to the least important for the company.

2) Determination of security objectives in this step, we set up a criterion for every asset identified in the previous step such that every asset has unique requirements, which requires security goals for every asset to be identified and are usually confidentiality, integrity and availability (CIA).

3) Risk analysis and assessment the third step is all about identifying the existing and potential risks that will violate any of the security goals, which will result in damaging the assets. After that, the degree of this risk is evaluated and measured by quantitative and qualitative analysis. The measurement and evaluation stop when the results are satisfying.

4) Risk treatment decisions regarding risks that have been measured and evaluated in the previous step are made in this step. There are four types of risk treatment: avoiding, reducing, transferring, or retaining risk [15].

5) Security requirements definition depending on the decision(s) made in the previous step and choosing the risk treatment type, the identification and derivation of the security requirements that work with the has been chosen to mitigate threats resulting from risks.

6) Control selection and implementation this is the last step of the process, in which, the implementation of the decisions made regarding mitigating and controlling risks and enhancing the information security level in the company through implementing countermeasures.

4.3. Security Modelling Languages

4.3.1. KAOS for Security

This section addresses KAOS for security as well as artifact security threat (*anti-goal element*). For security requirements analysis and elaboration by the use Goals KAOS element, the goal notion allows the expression of security requirements patterns in terms of anti-goals notion and vulnerabilities of the system that is being studied. These patterns can also include a definition of the solution, or counter measure, to the attack in terms of goals that avoid a given vulnerability.

4.3.2. Alignment between KAOS and ISSRM Domain Model

It was not flexible to build a model for *anti-requirements* or *anti-goal* using obstacles. We should have converted these obstacles into *operationalise* model by using the requirements element and pairing them with agents and finally adding operation elements.

Concepts related to ISSRM asset are represented by KAOS goal, requirement and expectations. Operation and object are used to present asset while security criteria are represented by goal and object attributes. Threat agent is presented by anti-agent while action method is represented by *operationalisation*, domain

and required conditions and operation. Vulnerability is defined by the domain property. At a higher abstraction level, anti-goal represents event while it represents threat in lower levels (in combination with anti-requirements and anti-expectation). Security requirement is represented by security goal. This goal can be refined further by security requirement and expectation [15]. Discussion about the name of the concepts is included in the ISSRM model.

Details on harmonisation between the concepts of KAOS and the domain model ISSRM (**Table 4**) can be found in [15] on three levels: Asset-related concepts, Risk-related concepts, Risk treatment-related concepts in details.

5. Safety and Security Engineering

We will address the domain model Safety and Security Engineering (S*a*S), is the result of the main contribution in integrating Section 3 and 4. We addressed the S*a*S domain produced followed by hazard/risk management process, KAOS-S*a*S modelling languages and running example *@RemoteSurgery* (**Appendix 1**). The example is run on S*a*S domain, the alignment between KAOS and S*a*S. The results of this process are discussed in section 6.

5.1. Common Method to Define Security and Safety (*SaS*) Domain Model

We will address the Security and Safety (S*a*S) domain produced followed by hazard/risk management process and KAOS-S*a*S modeling languages.

S*a*S is a requirement driven software engineering approach is a result of adapting the domain ISRM, which is also a risk analysis approach that inherited

Table 4. Concepts alignment between KAOS extended to security and the ISSRM domain model.

ISSRM domain model		KAOS extended to security	
		Synonyms in [21]	Language concept (modeling construct)
Asset-related concepts	Asset	Asset	Goal, Requirement, Expectation, Operation, Object
	Business asset		
	IS asset		
	Security criteria	Security Goal	Goal
Risk-related concepts	Risk	/	/
	Impact	/	/
	Event	Threat Obstacle; anti-goal	Goal, Requirement, Expectation (in anti-model)
	Threat		
	Vulnerability	Vulnerability, domain property	Domain property
	Threat agent	Attackers, malicious agent, anti-agent	Agent
	Attack method	Potential capabilities of the attacker	Operationalisation + Domain and required conditions + Operations
Risk treatment -related concepts	Risk treatment	Countermeasures	/
	Security requirements	Security goal, security requirement, security expectation	Goal, Requirement, Expectation
	Control	/	New model implementing security components.

the same method from the ISSRM domain model that deals with security and safety requirements.

S*a*S needs a requirements elaboration method and a design elaboration method in order to cover all the stages of development until implementation is obtained. S*a*S employs the KAOS for eliciting, modeling and analyzing security requirements and safety requirements while it employs the semi-formal specification language and temporal logic (LTL) formal specification language for deriving requirements specifications for both safety and security.

The idea behind integrating ISSRM and ISRM domain model is the main objective that is achieving a certain degree of dependability in the system-to-be. The thrive for achieving dependability in a system-to-be is because the principle of dependability deals with both intentional and unintentional incidents.

This model is the result of merging the ISSRM domain that focuses on security and the ISRM domain that focuses on safety producing a S*a*S domain model (Figure 7).

Both these domains have been addressed previously. Discussions of the safety and security (S*a*S) model (Figure 7).

1) Assets, anything that has value to the organisation and is necessary for achieving its objectives. These assets differ from a company to another whether it's (a.k.a. software, IT infrastructure, users, or strategic plan, etc.).

2) Control, in ISRM [5] and ISSRM models [15], we find that both models agree that there is control that is responsible for meeting the safety and security

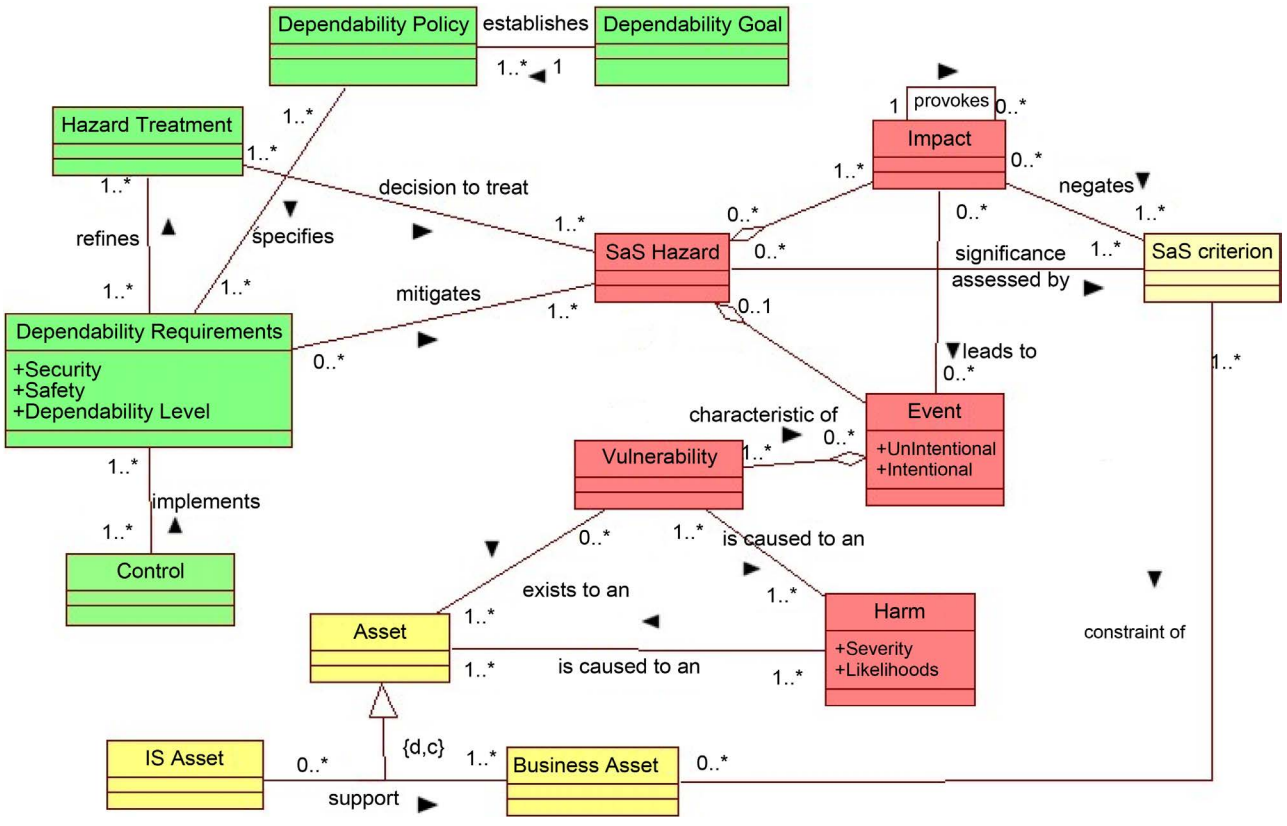

Figure 7. S*a*S domain model.

requirements and minimising the number of vulnerabilities by implementing safeguard and fail-safe methods.

3) SaS Hazard, concept is often used when dealing with systems that if an error occured; the environment in which the system exists would be affected. The researchers [23] have derived concept for it to describe both ISRM and ISSRM.

4) Event, is the combination of a threat ISSRM/hazard ISRM and one or more intentional/unintentional vulnerabilities. Intentional is feature of security engineering. There is always a motive and the intention behind planning an attack against the confidentiality, integrity and availability (CIA) concepts. Unintentional is feature of safety engineering. We do not know what accidents we will be facing and so, safety engineering follows very strict mathematical, qualitative, and quantitative methods to accurately analyse risks in order to limit the occurrence of any hazard and the control them.

5) Harm, is a significant damage, usually associated with an asset that is caused by a hazard, it come from the combination of identified severities and identified likelihoods.

6) Impact, the latent negative consequence of a hazard, where it negates dependability requirement criteria (Table 3).

7) Dependability requirements is the umbrella under which come many attributes including those of safety and security. These attributes are chosen based on the nature of the system to be developed and only one either for safety or security might be chosen, or both, according to what the researchers [4] have addressed in details the concepts and definitions of dependable. Dependability requirement should to be resilience to Intentional threats and Unintentional hazards.

8) Hazard Treatment is the kinds of quality requirements of dependability requirement after choosing with the attributes associated quality characteristics and quality measures. Standards must be defined according to the system to be developed. From a security perspective, these standards should be CIA standards, or in some cases non-repudiation ones. On the other hand, survivability, quality of service, fault tolerance, correctness, reliability, verification, validation, and maintainability from the safety perspective. It is important to mention that, depending on chosen dependability requirement, the dependability attributes (Table 3) and (Table 5) show a sub-criterion of performance attribute. The researcher Firesmith has listed the concepts of security, safety, and survivability under a more general concept, defensibility [5], which is a special case of dependability.

Table 5. Sub-criterion of performance [20].

Sub-criterion	Definition
Jitter	*Is the precision (i.e., variability) of the time when one or more events occur.*
Latency	*Is the time it takes to actually provide a requested service or allow access to a resource.*
Response time	*Time is the degree to which the time it takes to initially respond to a request for a service or to access a resource.*
Scheduleability	*is the degree to which events and behaviors can be scheduled.*
Throughput	*is the number of times that a service is provided within a specified unit of time.*

9) Dependability policy, are responsible for preventing chosen attributes requirements conflict as it determines the priority for each one.

10) Dependability goal, the operational level of the system is determined to work in the environment it was built for depending on whether this system will be used by everyone, professionals in a certain field, or a team that was well-trained. This is due to the fact that the product has a very high level of risk in case of human errors. After responsible authorities test the system, the final product of the system will be granted certificate.

After identifying the different terms used in each S*a*S source, our assumption that the concepts in the S*a*S model is not unified has been validated. Many different terms are used to depict the same concept. More than a dozen of different names have been found for some concepts in (**Table 6**).

The process of extraction and concepts alignment (**Table 6**) based on ISRM domain model, ISSRM domain model and S*a*S domain model and the definitions used for each concept in the S*a*S model.

5.2. S*a*S Risk Management Process

A lot of effort was invested in developing a common cross-industry approach to managing risk such as quality risk management [25] an recently, ISO 14971 [42], which defines the analysis requirements for medical devices that have the ability to connect to the network from the start of the production process for these devices.

We elicited these six steps [25] [42] process (**Figure 4**) for S*a*S risk management from the safety perspective through adapted ISRM model [5] and security perspective through ISSRM domain model [15]. The following Steps are (1 to 6) Summarised are follows.

1) Scope and context asset identification the process of searching for

Table 6. Concepts alignment between ISSRM, ISRM and S*a*S models.

Type	ISRM model	ISSRM model	S*a*S model
Asset-related concepts	Asset; Systems; Environment;	Asset; Business asset; IS asset;	Asset; Business asset; IS asset;
	Safety criteria; Quality sub-factor	Security criteria	SaS criteria
Risk-related concepts	Risk; Safety risk; Safety vulnerability; Danger; Hazard; Accident;	Risk; Impact; Event; Threat; Vulnerability; Threat agent; Attack method	Unintentional; Intentional; Hazard; Impact; Event; Vulnerability; Harm
Risk treatment-related concepts	Safety Goal; Safety requirement; Safety mechanism; Safeguard; Safety tactic	Risk treatment ; Security requirements; Control	Hazard treatment; Dependability Requirement; Dependability Goal; Dependability Policy; Control

stakeholders to address the safety and security implications, at the system level and their environments (a.k.a. assets, physical, social) for the purpose of defining the scope. After that, the assets of value for the company as well as the assets related to safety and security engineering need to be identified. The output of this step is the definition of the scope and its relation to the system and the environment and a priority list and rankings of assets to be secured from a safety and security perspective starting with the assets of the highest priority.

2) Determination of dependability objectives in this step, we set a dependability need for every asset identified in the previous step, while each asset has its own characteristics, which requires the identification of Safety/Security goals for each of these assets as summarized in (**Table 3**). The existing attributes in **Table 3** give a general idea on the attributes of critical systems. However, its not necessary that each system contain each attribute keeping in mind that the more attributes there are in a system, the more it would cost. What is (**Table 3**) and (**Table 5**) is for illustration purposes only. The full sets of attributes are available at [4] [20].

3) Risk analysis and assessment the third step consists of the identification of existing and potential hazards that are likely to violate the safety/security goals resulting in accidents. Without doubt, these accidents will cause damage to assets and environment.

Using hazard analysis tools such as HAZOP, FTA, AT, FMEA for instance, and using the Scenario, AF, KAOS and Misuse-Cases from a security perspective. After identification, these hazards are evaluated and the degree of risk is measured using quantitative and qualitative analysis. At this stage, the defining the likelihood of occurrence, defining consequence categories, and risk matrix are produced and the result is full information on these hazards. After that, ALARP principle is implemented to measure the tolerance of each hazard [17]. If the results are dissatisfying, the entire process has to be performed again starting from step 1, otherwise, the process proceeds to step 4.

4) Risk treatment in this step the decision is made regarding these hazards. These types of risk treatments are divided to three categories: prevention, reducing, or retaining risk.

5) Dependability requirements definition depending on the decision(s) made and choosing the measures in the previous step, we derive the Control, and the strategic decision that will satisfy dependability requirement to define safety integrity level (SIL) target that complies with what has been chosen in order to mitigate and control harms resulting from hazards.

6) Constraint selection and implementation in this step, the decisions made regarding hazards are implemented by setting constraints that comply to SIL target in parallel with implementing safeguards for intentional and unintentional hazards. To ensure the compatibility of the chosen dependability criterion for each asset individually by referring to the dependability policy.

Safety-critical and security-critical software systems are dynamic and interactive resulting in having unintentional hazards. The upgrading process is conti-

nuous as the main objective of monitor the residual risk and its compliance to the standards and certificate [16] [18].

5.3. Safety and Security Modeling Language

5.3.1. KAOS for S*a*S

This section addresses KAOS for safety and security as well as artifact S*a*S Obstacles (Hazard and Threat).

For safety-critical requirements analysis, it is crucial to deal with obstacles KAOS element. The Obstacles element is a common element between safety and security (Table 7).

Obstacles KAOS element are not only limited to representing safety goals but also depends mainly on the system-to-be, its specifications and specific environment (Table 7). It is possible to deal with inaccuracy obstacles or non-satisfaction obstacles [9]. Knowing the classification domain; *first step from SaS Risk Management Process*—(*a*) *Scope and context asset identification*, of obstacles enables and enhances finding suitable treatments.

The **Running example** *@RemoteSurgery* that was mentioned in Appendix 1 through S*a*S risk management process introduced in section 5.2 containing six steps and implements them on the example using KAOS modelling language.

1) Scope and context asset identification

This step is done through the definition of goals and their refinement in the KAOS safety and security goal model, as depicted in (Figure 8) The main goal studied in the example is *Maintain* [*Correctness Movement Scale*] for both Safety and Security modelling analysis, which is refined in the context domain property *Surgeon Well Trained* and the sub-goals from safety side *Quality Of Image* associated to the agent *Camara and* sub-goals from security side *Minimal Latency* associated to the agent *Service Provider*.

More details about the IS are given in the *Safety* operation model *Quality Of Image*. The goal *Quality Of Image* is associated to the agent *Camara.* It also performs other operations (*Boundary Detection and Image Acuisitions*). And More details about the IS are given in the *Security* operation model *Minimal Latency*. The goal *Minimal Latency* is associated to the agent *Service Provider.* He also performs other operations (offer *High-Bandwidth* communication) (Figure 8).

2) Determination of dependability objectives

Figure 8, the determination of S*a*S objectives is done in the same model and

Table 7. Obstacle categories, adapted from [21].

Types of Obstacle	To Represent
Hazard Obstacle	Goal Safety
Threat Obstacle	Security Goal
Dissatisfaction Obstacle	Satisfaction Goal
Misinformation Obstacle	Information Safety
Inaccuracy Obstacle	Accuracy Goal
Unusability Obstacle	Usability Goal

generally in the same time as the elicitation of other goals. *Minimal Latency* and *Quality Of Image* are an example of S*a*S objective; S*a*S need, meaning that we need the accuracy, correctness, robustness, integrity and availability of *Movement Scale*; *Surgical Maneuvers*; *Minimal Latency*; *Insertion Of Malicious Software*; Redundancy Components.

3) Risk analysis and assessment

We elaborate safety and security requirements by negating the S*a*S goal *Correctness Movement Scale* (for Security and Safety goals) to obtain the root Obstacles *No Availability* (Figure 8). We elaborate by hazard analysis to refine the main Obstacle *No Availability* to three sub-obstacles; *Driver Unresponsive*; *Communication Under DDOS Attack*; *Hardware Unresponsive* (Figure 9).

4) Risk treatment

Hazard treatment is defined through the countermeasure chosen for handling the *Safety and Security Obstacle*, and its associated vulnerabilities, obstacles (Figure 9). In our example, the countermeasure chosen to *prevent hazards*; controlling and interacting with hazards so they do not become accidents.

5) Dependability requirements definition

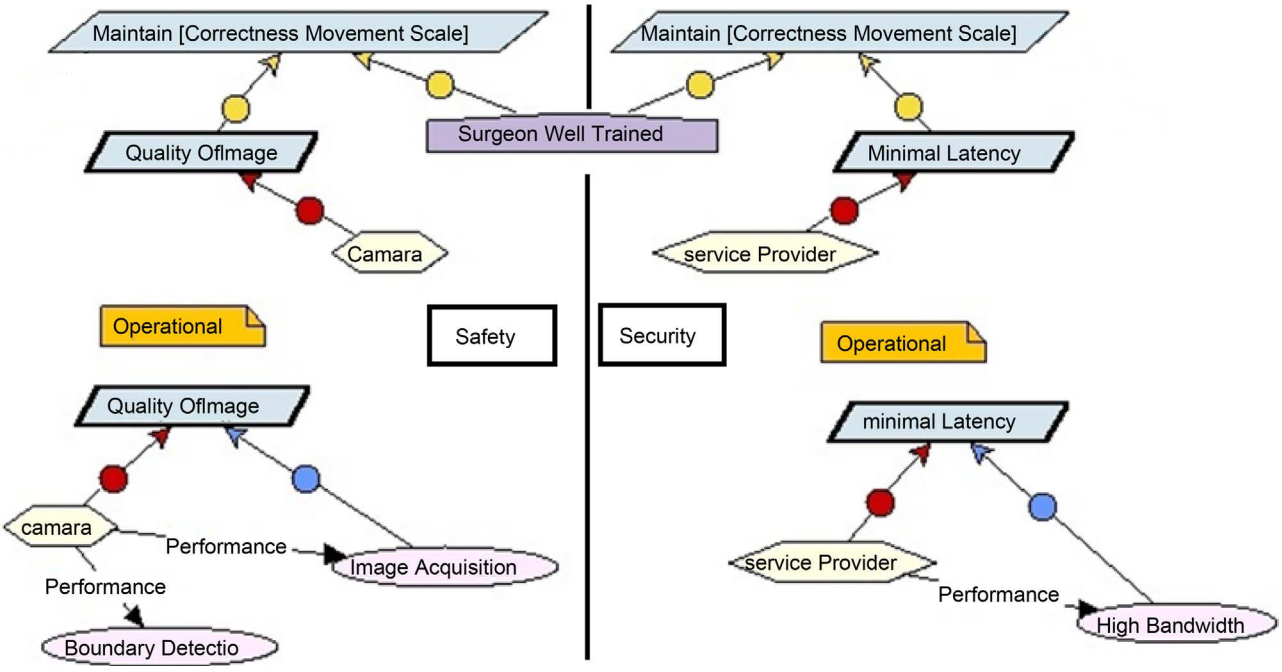

Figure 8. The same asset for safety and security, objective modelling in KAOS.

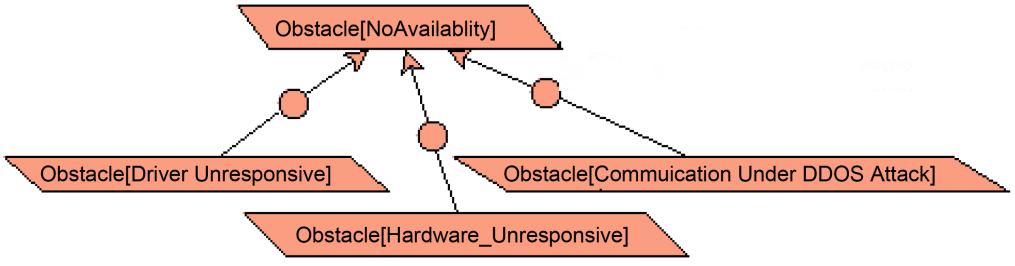

Figure 9. Safety and security obstacle hazard analysis.

Obstacles prevention by introduce a new goals Avoid [*High Latency*]; *Achieve* [*Redundancy Components*]; Avoid [*Failed Software*]; Avoid [*Insertion Of Malicious Software*] as a countermeasures. *High Latency* and *Redundancy Components* goals refined to into two requirements *Redundancy Communication Line* and *Minimal_Latency*, both requirements are assigned to the *Service Provider* agent. *Failed Software* and *Insertion Of Malicious Software* goals refined to into two requirements *Driver responsive* and *Watchdog Check*, both requirements are assigned to the *Computer Software* agent (Figure 10).

6) Constraint selection and implementation

The update of the safety goal model, which might include the refinement and the *operationalisation* of the new added *Avoid* and *achieve* goals to meet our expectations, constitutes the new system-to-be, as in (Figure 10).

5.3.2. Alignment between KAOS and S*a*S Domain Model

In this section we will contribute towards Alignment between KAOS and S*a*S and create relationship and mapping between the concepts of both KAOS and S*a*S. Discussion about the name of the concepts included in the S*a*S model.

After identifying the different terms used in each S*a*S source, our assumption that the terminology in the S*a*S model is not unified has been validated. Many different terms are used to depict the same concept. More than a dozen of different names have been found for some concepts in (Table 6).

After identifying the concepts comes the aligning process (Table 8), and define relationships between KAOS and S*a*S. Discussion about the alignment tables.

Asset-related concepts, KAOS is mainly focused on the security of the system-to-be, but it does not make a separation between the IS and business aspects. Thus, we align the Asset S*a*S concepts concerning assets with the KAOS Strategic goal, requirement and expectation (Table 8). Moreover, their *operationalisation* in operation and object are also assets. In KAOS, states of the system-to-be are described using object attributes. The purpose of the Safety goals is to achieving a target level of safety or one of its sub-factors. In terms of KAOS, this means that the safety and security goals should define S*a*S criterion

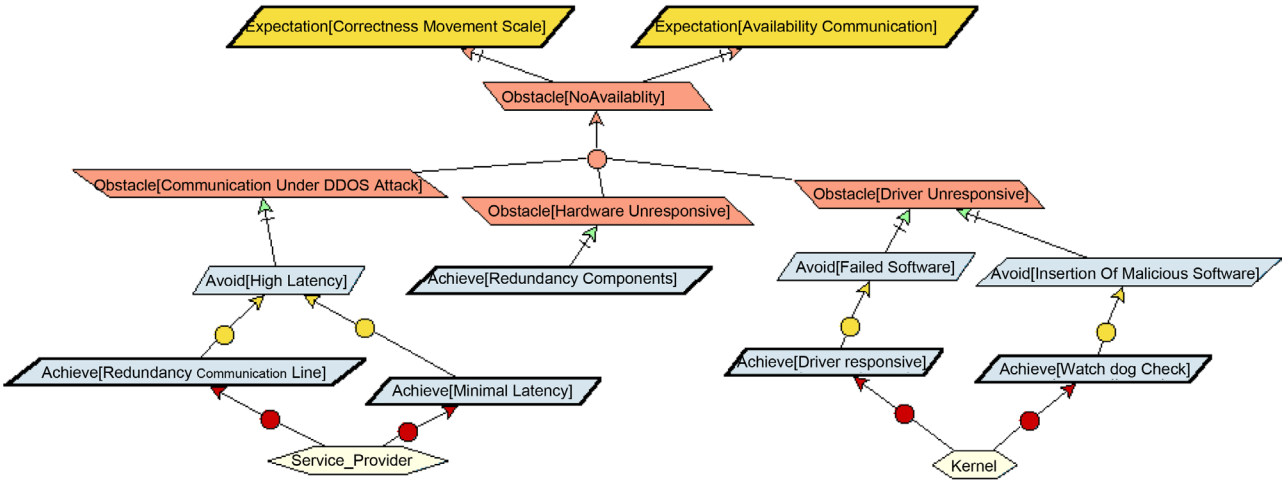

Figure 10. Safety and security requirements and control modelling in KAOS.

Table 8. Concept alignment between KAOS extended to S*a*S.

	SaS Model	KAOS extended to SaS	
		Synonyms in [21].	Language concept (modelling construct)
Asset-related concepts	IS Asset	Asset	Business Goals Strategical Goals, Requirement, Expectation,
	Business Asset		
	SaS criterion	/	Goal verbs, Object attribute
	Impact	/	/
	Harm	/	/
Risk-related concepts	SaS Hazard	Hazard obstacle; Threat obstacle; Dissatisfaction obstacle; Misinformation obstacle; Inaccuracy obstacle; Unusability obstacle	Goal; Goal Safety; Security Goal; Satisfaction Goal; Accuracy Goal; Usability Goal; Requirement; Expectation (in anti-model).
	Vulnerability	Vulnerability, domain property	Domain property
	Event; Unintentional; Intentional	Attackers; malicious agent; non-malicious agent; anti-agent	Agent; Operationaisation
Risk treatment-related concepts	Hazard treatment; Dependability Goal	Countermeasures	/
	Dependability Requirement; Dependability Policy	Safety-goal, Security-goal, security requirement, security expectation	Goal, Requirement, Expectation
	Control	/	New model implementing SaS components.

(Table 3), which are concerned by potential risk events and hazard and/or threat (Table 7) [21]. Thus, we align both (*safety*) goals concerned by *Expectation*; *anti-requirements*; *anti-goals* with S*a*S requirement criteria.

Risk-related concepts, In Table 8, we align together S*a*S (*unintentional and intentional*) hazard (Table 7) with KAOS Obstacle, Negative scenarios (also called hazard obstacle; threat obstacle; dissatisfaction obstacle; misinformation obstacle; inaccuracy obstacle; unusability obstacle). *Obstacle* can be identified at various abstraction levels, so they might need to be refined until they become anti-requirements or anti-expectations (assigned to an anti-agent). At higher abstraction levels, an anti-model might be considered as the event, which, according to the S*a*S model, is a combination of a hazard and one or more vulnerabilities (safety or security or both or see Table 7). At lower abstraction levels, an anti-model (*anti-requirement or anti-expectation*) is a hazard, which is an unintentional attack or intentional to assets. The language concepts for *anti-model*, *anti-requirement* and *anti-expectation* remain respectively goal, requirement and expectation.

In Table 8, we align S*a*S Vulnerability and the KAOS domain property. The KAOS domain property is a hypothesis about the domain that holds indepen-

dently of the system-to-be. In correspondence, S*a*S vulnerability is defined as attributes of assets. Following the S*a*S model, Hazards (**Table 7**) cause harm to the assets, due to an unintentional accident when dealing with safety engineering, is due to an intentional attack when dealing with security engineering.

In KAOS, an anti-agent monitors or controls objects and their attributes, and is thereby capable to hazardous the system-to-be. In (**Table 8**), we align S*a*S unintentional and intentional and KAOS malicious agent; non-malicious; anti-agent. The S*a*S model is not clear is there attack method characterises the means by which intentional or/and unintentional attacker carries out the attack. In KAOS an *anti-agent* performs operations that satisfy an anti-model. Operations change the state of the system-to-be using input/output relationships over the objects and their attributes. This means that by performing operations, the anti-agent (malicious agent; non-malicious) breaks the safety and security criteria (**Table 3**) (related to object attributes). (**Table 8**), we align S*a*S unintentional and intentional with the KAOS constructs used to *operationalise* the *anti-model*, namely *operationalisation*, domain and required conditions and operation.

Risk treatment-related concepts, S*a*S hazard treatment corresponds to the countermeasures [19] [21] that are elaborated after identification of the anti-model. Countermeasures are not KAOS modeling concepts, but rather modeling idioms or patterns adopted by modelers. In KAOS, the countermeasures usually result in new dependability goals, which need to be refined further into realisable safety and security requirements and expectations. In (**Table 8**), we align S*a*S security requirements; safety requirement and the KAOS Safety goal; security goals (*requirements and expectations*). The refinement and *operationalisation* of the new safety and security goals, their concerned objects and attributes, and their assignment to agents, lead to new system-to-be components realising the necessary safety and security means. With respect to the ISSRM and ISRM information models, these new system components correspond to Control.

6. Validation

In this section, consists of the validation and comparison between the uses of KAOS in running example on the suggested S*a*S domain and the use of Systems-Theoretic Accident Model and Processes (STAMP; **Appendix 2**) techniques languages (STPA; *STPA-sec*; supplementary material section) running the same example on S*a*S domain (section 5).

We address the results achieved from this research. Goal, Question, Metric approach (GQM) [27] will be used in questioning the metrics used for validating the level of maturity of the S*a*S domain model. In other words, we want to count the concepts this domain model inherited from both the ISSRM [15] and the ISRM [5] domains to serve the application of the dependability concept. The metric here is concept completeness.

The second goal is divided into two parts; since S*a*S domain model has inherited several concepts from the ISSRM and the ISRM domain models that over-

lap, which affected the result of the alignment process between S*a*S concepts and KAOS patterns elements. This is because KAOS does not support constructing redundant elements [28]. The metric here is semantic completeness. The second part of this goal is whether the alignment process between KAOS verbs concepts and S*a*S concepts is semantic correctness.

6.1. Case Study

To sum it up, a S*a*S information model has been created (section 5) and used one of the Goal modeling language (GML) languages, namely KAOS modelling language (chapter 6) to implement the hazard management process for S*a*S on the example *@RunningSurgery*, and the alignment process between the KAOS and the S*a*S domains (section 5).

Furthermore, we used STAMP and its STPA-sec modelling language, which are categorised under scenario-based approach. We first align S*a*S and STAMP using the example *@RunningSugery* using a *Scenario-based approach*, and then we applied hazard management process from the safety side using STPA on the same example used in S*a*S. The hazard management process was applied from the security side using *STPA-sec*.

6.2. Discussion

We discuss the aforementioned goals. To avoid repeating tables and figures, we will refer to them when necessary.

In **Table 6**, we extracted the concepts of the ISSRM, ISRM, and S*a*S domains then dividing these concepts into three categories; *Asset-related concepts*; *Risk-related concepts*; *Risk treatment-related concepts*. We find that it is clear that the concepts of S*a*S domain are Risk-related concepts are redundant.

The second goal consists of two parts; in (**Table 7**), we see that obstacles are divided into six categories. When represented using KAOS, these six categories are reduced to one type that is later built and can be customized as to which obstacles category it belongs to by adding annotations. Secondly, there were clear indications that it affected semantic correctness during the alignment process between KAOS elements and S*a*S concepts (**Table 8**). This is due to the fact that each KAOS patterns (a.k.a. *Avoid*, *Maintain*, *and Achieve*) are met by more than one concept from the S*a*S domain. This applies to the obstacles element since there are six categories (**Table 7**) [21]. For that, we intended to leave it unexplained using a single term {*Obstacles*} and used an explicit term *Threat Obstacle* that deals with security instead. On the other hand, *Hazard Obstacles* deals with the safety perspective to support semantic KAOS-S*a*S in our work.

We recall section 5.3 and the supplementary material. The results contain differences. These differences are due to two reasons: The example *@RunningSurgery* was run using KAOS, which is considered from the GML category. KAOS patterns (a.k.a. Avoid, Maintain, and Achieve) were used to create obstacle models, and derive milestone from it.

6.3. Cases

We compare the results we got from applying KAOS-S*a*S with the results we have from applying the example *@RunningSurgery* and measure the degree of likelihood using STAMP (*STPA*; *STPA-sec*).

The same example *@RunningSurgery* was run using *STPA* and *STPA-sec*, which are considered from the scenario-based category that use textual description of the analysis process. Furthermore, the approaches used in the hazard analysis for security (*STPA-sec*) do not differ much from the approaches used in threat analysis for safety. This is clear in the phase "*Identifying unsafe/unsecure control actions*" (**Table 9**) STPA-sec as security and safety are inseparable as mentioned by Young [29] [30].

This is due to the fact that *STPA-sec* does not take into account traditional security standards like confidentiality, integrity and availability, which leads to ambiguity around the standards that to be used when running the example from the security side using *STPA-sec*. however, it has the advantage of being scenario-based because it was helpful using textual description.

6.4. Lesson Learn

Hazard/Risk management process was the entry point for the integration process as the interface interplays between safety requirements and security requirements from the aspect of system functionality and what the system should and should not do. S*a*S is the result in integrating ISRM hazard management and ISSRM risk management were aligned of between KAOS and S*a*S domain model. It became possible to analysis safety and security in a consistent and using one modeling language tool.

Finally, we would like to conclude that using KAOS in this research was suitable to run the experimental researches, and easy to learn as supported by the study conducted by [22]. Goal modeling tool, *Objectiver*, was used in the creation of goal models to give contextualization to these goals, and is rich in element shapes [31] that supported the use of KAOS.

7. Conclusion and Further Work

Integrating safety and security should occur during the initial development phases of the system because it is a very important step for safety and security

Table 9. Summary of steps risk/hazard management process for S*a*S, STPA and STPA-*sec*.

S*a*S	STPA	STPA-sec
1)-Scope and context asset identification	1)-Identify accidents and hazard	1)-Determining unacceptable losses
2)-Determination of dependability objectives	2)-Construct functional control structure	2)-Creating a model of the high level control structure-HLCS
3)-Risk analysis and assessment	3)-Identify unsafe control actions	3)-Identifying unsafe/unsecure control actions
4)-Risk treatment	4)-Identify causal factors and control flaws	4)-Developing security requirements and constraints
5)-Dependability requirements definition		5)-Identifying casual scenarios
6)-Constraint selection and implementation		

engineers in order to discover the causes of hazard and fault. Furthermore, it is a very important because it's the only thing that covers the gap between safety engineers and security engineers especially since a security engineer knows the risks a system could face and therefore has to protect the system and the equipment from any threats. On the other hand, a safety engineer does not know what the hazard would be or their effect on the environment. Therefore, a safety engineer will have to discover the unknown hazard the system could possibly face. This is the critical point from which hazard and risks are derived by both types of engineers using a systematic approach to communicate.

We investigated the available information domain models for safety engineering and found a domain model ISRM that addresses safety engineering, which has enriched the understanding of the concepts used in risk management process from the safety engineering aspect. Furthermore, we found the ISSRM domain model from the security engineering aspect, which has enriched the understanding of the concepts used in risk management process. Therefore, we performed an alignment between KAOS and ISRM domain model concepts. We performed an alignment between KAOS and ISSRM domain concepts. This has resulted in extended coverage for the concepts resulting from the integration between safety engineering and security engineering in the risk management process and enlisting all of them in a table.

We used KAOS that is classified under goal-oriented languages. We have found that KAOS enables a representation method for security and safety hazard/risk management together by used obstacles method.

We propose conceptual framework through the creation of S*a*S information domain model that integrates safety and security domains giving a better opportunity for interplay and integration to find a middle ground between the ISRM hazard management and ISSRM risk management as well as unifying definitions through their mappings onto the common concepts. We performed an alignment between S*a*S domain model concepts and KAOS concepts elements.

We have investigated alignment between the S*a*S domain models for hazard/risk management with the modeling language KAOS, which has given the possibility for a better method to derive safety and security requirements in early stages from the beginning of the system development life cycle by used *Obstacles* approach. The alignment between S*a*S domain model and KAOS enhances the cooperation and facilitates communication and interaction between stakeholders.

Our future work will concentrate on more exploring and using formal and semi-formal language for S*a*S. KAOS supports using semi-formal and linear formal specification language (LTL) to describe *Goals*, *Obstacles* and to perform logical proofs, which gives accuracy and reveals ambiguities. This is what sensitive and critical systems are in need for, which integrates between safety and security after identifying the requirements specifications of both and later reduced to formal languages that reveals complications resulted from achieving the goals of safety and security. Formal specifications can assist in correcting design of system requirements specifications and improve the quality of system-to-be.

References

[1] Committee on National Security Systems (CNSS). National Information Assurance (IA) Glossary (CNSS Instruction No.4009). Committee on National Security Systems (CNSS), National Security Agency (NSA), Fort Meade, May 2003.

[2] Piètre-Cambacédès, L. and Chaudet, C. (2010) The SEMA Referential Framework: Avoiding Ambiguities in the Terms "Security" and "Safety". *International Journal of Critical Infrastructure Protection*, **3**, 55-66. https://doi.org/10.1016/j.ijcip.2010.06.003

[3] Benjamin, F., Seda, G., Maritta, H., Thomas, S. and Holger, S. (2010) A Comparison of Security Requirements Engineering Methods. *Requirements Engineering*, **15**, 7-40.

[4] Avizienis, A., Laprie, J.-C., Randell, B. and Landwehr, C. (2004) Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, **1**, 11-33. https://doi.org/10.1109/TDSC.2004.2

[5] Firesmith, D.G. (2003) Common Concepts Underlying Safety Security and Survivability Engineering. (No. CMU/SEI-2003-TN-033) Software Engineering Institute, Carnegie Mellon University.

[6] Firesmith, D.G. (2004) A Taxonomy of Safety-Related Requirements. Position Paper at the Requirements for High Assurance Systems (RHAS) Workshop at the 12th IEEE International Conference on Requirements Engineering (RE'2004) in Kyoto, Japan on 6 September 2004, 11 p.

[7] Firesmith, D.G. (2012) Engineering Safety- and Security-Related Requirements for Software-Intensive Systems. *The* 11*th IASTED International Conference on Software Engineering* (*SE* 2012) *in Crete*, Greece on 18 June 2012.

[8] Mayer, N., Rifaut, A. and Dubois, E. (2005) Towards a Risk-Based Security Requirements Engineering Framework. *Proceedings of the* 11*th International Workshop on Requirements Engineering*: *Foundation for Software Quality* (*REFSQ*05), in Conjunction with the 17th Conference on Advanced Information Systems Engineering (CAiSE'05).

[9] Yu, E.S.K. (1997) Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering. *RE* '97: *Proceedings of the* 3*rd IEEE International Symposium on Requirements Engineering*. *IEEE Computer Society*, Washington DC, 226.

[10] Yu, E.S.K. and Liu, L. (2001) Modelling Trust for System Design Using the i * Strategic Actors Framework. *Proceedings of the Workshop on Deception*, *fraud*, *and Trust in Agent Societies Held during the Autonomous Agents Conference*, Springer, London, 175-194.

[11] Liu, L., Yu, E. and Mylopoulos, J. (2003) Security and Privacy Requirements Analysis within a Social Setting. *Proceedings of* 11*th IEEE Requirements Engineering Conference*. IEEE Press, 151-161. https://doi.org/10.1109/icre.2003.1232746

[12] Piètre-Cambacédès, L. and Bouissou, M. (2010) Modeling Safety and Security Interdependencies with BDMP (Boolean logic Driven Markov Processes). 2010 *IEEE International Conference on Systems Man and Cybernetics* (*SMC*), 10-13 October 2010, 2852, 2861.

[13] Kriaa, S., Bouissou, M. and Piètre-Cambacédès, L. (2012) Modeling the Stuxnet Attack with BDMP: Towards More Formal Risk Assessments. 2012 7*th International Conference on Risk and Security of Internet and Systems* (*CRiSIS*), 10-12 October 2012, 1-8.

[14] Firesmith, D. (2010) Engineering Safety- and Security-Related Requirements for Soft-Ware-Intensive Systems: Tutorial Summary. *Proceedings of the* 32*nd ACM/IEEE International Conference on Software Engineering*, Volume 2 (ICSE '10), Vol.

2. ACM, New York, 489-490.

[15] Mayer, N. (2009) Model-Based Management of Information System Security Risk. Doctoral Dissertation, University of Namur, Namur.

[16] C. Warren, A. (2012) Engineering Safe and Secure Software Systems. 1st Edition, Artech House, Boston.

[17] Redmill, F. and Consultancy, R. (1999) An Introduction to the Safety Standard IEC 61508. *Hazard Prevention*, **35**, 20-25.

[18] Brazendale, J. (1995) IEC 1508: Functional Safety: Safety-Related Systems. In Software Engineering Standards Symposium, August 1995. (ISESS'95)'Experience and Practice', Proceedings, Second IEEE International, IEEE, 8-17.

[19] Van Lamsweerde, A. and Letier, E. (2000) Handling Obstacles in Goal-Oriented Requirements Engineering. *IEEE Transactions on Software Engineering*, **26**, 978-1005. https://doi.org/10.1109/32.879820

[20] Firesmith, D.G. (2007) Engineering Safety and Security Related Requirements for Soft-Ware Intensive Systems. 29*th International Conference on Software Engineering-Companion*, 2007. ICSE 2007 Companion, 169-169.

[21] Van Lamsweerde, A. (2009) Requirements Engineering: From System Goals to UML Models to Software Specifications.

[22] Matulevičius, R. and Heymans, P. (2007) Comparing goal Modelling Languages: An Experiment. In: Sawyer, P., Paech, B. andHeymans, P., Eds., *Proceedings of the* 13*th International Working Conference on Requirements Engineering: Foundation for Software Quality* (REFSQ'07), Springer-Verlag, Berlin, Heidelberg, 18-32.

[23] Sadvandi, S., Chapon N. and Pietre-Cambacedes, L. (2011) Towards a System Engineering Approach to Master Safety and Security Interdependencies. *Proceedings of the* 23*rd International Conference on Software & Systems Engineering and Their Application* (*ICSSEA*), Paris.

[24] Romani, M.A.S., Lahoz, C.H.N. and Yano, E.T. (2009) Dependability Attributes for Space Computer Systems. P*roceedings of* 3*rd CTA-DLR Brazilian Symposium on Aerospace Engineering and Applications/ Workshop on Data Analysis and Flight Control*.

[25] Guideline, I.H.T. (2005) Quality Risk Management. Q9, Current Step, 4, 408.

[26] Hollnagel, E., Woods, D.D. and Leveson, N., Eds. (2007) Resilience Engineering: Concepts and Precepts. Ashgate Publishing, Ltd., Ashgate.

[27] Basili, V., Caldiera, G. and Rombach, H.D. (1994) The Goal Question Metric Approach. John Wiley & Sons, Inc.

[28] Matulevičius, R. and Heymans, P. (2007) Comparing Goal Modelling Languages: An Experiment. *Requirements Engineering: Foundation for Software Quality Lecture Notes in Computer Science*, **4542**, 18-32.

[29] Young, W. and Leveson, N.G. (2013) Systems Thinking for Safety and Security. ACSAC.

[30] Young, W. and Leveson, N.G. (2014) An Integrated Approach to Safety and Security Based on Systems Theory. *Communications of the ACM*, **57**, 31-35.

[31] Matulevičius, R., Heymans, P. and Sindre, G. (2006) Comparing Goal-Modelling Tools with the RE-Tool Evaluation Approach. *Information Technology and Control*, **35A**, 276-284.

[32] Jackson, D., Thomas, M. and Millett, L.I., Eds., Committee on Certifiably Dependable Software Systems, National Research Council (2007) Software for Dependable Systems: Sufficient Evidence? National Academy of Sciences, Washington DC.

[33] Fei, B.w., Ng, W.S., Chauhan, S. and Kwoh, C.K. (2001) The Safety Issues of Medical Robotics. *Reliability Engineering & System Safety*, **73**, 183-192.

[34] Marescaux, J., Lero, J., Rubino, F., Vix, M., Simone, M. and Mutter, D. (2002) Transcontinental Robot Assisted Remote Telesurgery: Feasibility and Potential Applications. *Annals of Surgery*, **235**, 487-492. https://doi.org/10.1097/00000658-200204000-00005

[35] Anvari, M., McKinley, C. and Stein, H. (2005) Establishment of the World's First Telerobotic Remote Surgical Service: For Provision of Advanced Laparoscopic Surgery in a Rural Community. *Annals of Surgery*, **241**, 460-464.

[36] Anvari, M. (2007) Remote Telepresence Surgery: The Canadian Experience. *Surgical Endoscopy*, **21**, 537-541.

[37] Arata, J., *et al.* (2006) A Remote Surgery Experiment between Japan-Korea Using the Minimally Invasive Surgical System. *Proceedings* 2006 *IEEE International Conference on Robotics and Automation*, ICRA 2006, IEEE, 2006.

[38] Hollnagel, E., Woods, D.D. and Leveson, N. (2006) Resilience Engineering Concepts and Precepts. Chapter 8: Engineering Resilience into Safety-Critical Systems. CRC Press. US, NW.

[39] Leveson, N.G. (2012) Engineering a Safer World: Systems Thinking Applied to Safety. The MIT Press, Cambridge, MA.

[40] Leveson, N.G. (2013) An STPA Primer Version 1. August 2013.

[41] ISO, E. (2009) 14971: 2009. Medical Devices-Application of Risk Management to Medical Devices (ISO 14971: 2007, Corrected version 2007-10-01). CEN/CENELEC, Brussels, Belgium.

[42] Council Directive 93/42/EEC of 14 June 1993 concerning medical devices, Official Journal L 169, 12/07/1993 P. 0001-0043.

Appendices

Appendix 1, this appendix describes the running example *@RemoteSurgery.*

Appendix 2, this appendix addresses the alignment between the concepts of STAMP Approach and S*a*S domain model and detailed explanation on the use of STAMP approach concept using the S*a*S domain model in running the example *@RemoteSurgery*.

Appendix 1

The motivation behind *@RemoteSurgery* example is because it has focus on both safety and security perspectives as each of which affects the other. The example from [32] "*In the summer of* 2005, *radiotherapy machines in Merseyside*, *England*, *and in Boston were attacked by computer viruses. It makes little sense to invest effort in ensuring the dependability of a system while ignoring the possibility of security vulnerabilities…*" are of when the devices are not connected to the network. There is a risk on the security side, which will have a negative effect on safety, which lead to death.

Furthermore, these advanced devices run on an operating system like URObot [33] that uses a Linux Red Hat distribution using Fast Light Tool Kit (FLTK) in the GUI among other things. We realize that these systems can be infected with viruses and compromised like the rest of the systems, which affects safety.

The following description is an extract from [34] Operation Lindbergh, Operation Canada Tele-Surgeries [35] [36], and experiment between Japan-Korea [Jumpei *et al*., 2006]. *@RemoteSurgery* consists of three main components; the **patient info** that will be shared, the **master console** that is located in the same operating theatre where the surgeon will be controlling the surgery on his side, and the slave console located next to the patient. The slave console received commands from the operating surgeon sent from the master console. These commands are then executed on the patient's body directly without any human interference. The third component is **telecommunications technology** used to link the master and **slave console** in order to transmit the video live feed to the operating surgeon and for the surgeon to send the operating commands to the slave console. Transmitting and receiving operations in this case are subject to packets loss, which puts the operation at risk, besides making sure transmission is not delayed.

Our goal is to represent hazards from the safety aspect and threats from the security aspect in a single domain model that integrates the two aspects and performs hazard and threat analysis using KAOS as addressed by the researchers in Operation Lindbergh [34], Operation Canada Tele-Surgeries [35] [36], experiments between Korea and Japan [37].

Appendix 2

Alignment between the concepts of STAMP Approach and S*a*S domain model

We address part of the description of *@RemoteSurgery* (**Appendix 1**) and re-

flect it on the STAMP domain model. Using the concepts in S*a*S information domain model (Section 5.1) here to explain STAMP; *Scenario-based* [29] [30] in a narrative way.

@RemoteSurgery, A surgeon that is well trained on using the console is considered one of the important assets. Also, the operation itself as well as the patient who will get the operation is done using this console. The operation will be performed in a customized environment that meets the standards for operations, which is the hospital and its assets. Without this environment, no operation can be performed without this environment and especially the operations room, which has the necessary tools and the console that the trained surgeon will use to perform the operation (**Table 10**).

From a safety criterion perspective, it is more about the tools and the environment that have to meet certain standards that comply with MDD, which are as follows: failure tolerance; correctness; accuracy; availability, and human backup element that comes from resilience engineering [38]. It is an important factor in this medical field as it is the surgeon that will make a decision and interact manually in case the system becomes out of control.

From a *security criterion* perspective, we are more concerned about the confidentiality of information since it is medical data and being confidential is the normal status. For that, we require CIA (**Table 3**) since confidentiality requires not revealing medical information and treatment costs.

MDD [42] has divided the medical instruments into four categories depending on the hazard level. The robotic medical instruments are classified under *Class IIb* as shown in (**Table 11**).

According to MDD, risk analysis and hazard identification must be performed in the design phase following the Drift Correction principle but since the security side is taken into consideration, hazard identification process must be a comprehensive one. The standard IEC 1508 [41] confirms performing that as well as SIL; Reliability-based identification.

In the phase of *dependability requirement* specification, the execution of both

Table 10. Potential summery about STAMP Asset.

Asset	Reflection
Organisational asset	Doctors, Operation, Patient
System asset	Surgery consoles, Network connection
Property	Surgery operating room
Environment	Hospital

Table 11. MDD divides devices into four classes of qualitative scales.

MDD class	Hazard Level
Class I	Low risk,
Class IIa	Medium risk
Class IIb	Medium risk
Class III	High risk

analytic and holistic process and using a different technique, each of the components that interact with the system to be analysed and the interaction with each of these components relying on other components that already exist in the system without separation as well as independent analysis of each of these components. These components are (a.k.a *hardware*, *software*, *humans*, *environment*), which gives us a better overview in dealing with hazards and treatment plans that work with the *dependability goal*.

The process of risk treatment is related to cost, which is the result of analyzing the *dependability goal* phase. There's an inverse relationship between *cost* and *safeguard* requires execution in the system-to-be. The estimated cost resulted from *quantitative* and *qualitative* analysis for both *safety* and *security* requirement (Leveson used term *Constraints*). Theoretically speaking, it is easy to do, but practically, it is very difficult to define the suitable safeguards that will be used with safety. For that, we have to keep into account the safety policy (this complies with [39] [40] "*Conflicts between goals and constraints can more easily be identified and resolved if they are distinguished*"), that will be used in the system-to-be. These policies are used to comply with the *dependability goal* requirements.

On the top of the component hierarchy pyramid for the STAMP model, we find congress and legislature, which controls and organizes government regulatory agencies; industry and user associations; insurance companies; unions; and courts. In fact, there are several cases regarding legally allowing the use of *@RemoteSurgery* in hospitals. Furthermore, insurance companies aren't into insuring patients who want to have their surgery performed using *@RemoteSurgery*. Similarly, industry associations are developing training curriculums to train surgeons on using *@RemoteSurgery* and give them tests to measure their abilities and certify them. User associations affect the patient acceptance or declining the use of *@RemoteSurgery*. STAMP takes into consideration in the socio-technical cases.

The outcome of this work will be compared to validation section.

Supplementary Material

Supplementary Material A: We will address the phases of the *STPA* process from the safety perspective, and *STPA-sec* process from the security perspective. Web link for the technical report here https://goo.gl/GKaoai