Vulnerability Analysis Approach To Capturing Information System Safety Threats and Requirements

Oluwasefunmi, Arogundade [1] ; Adio, Akinwale [2] ; Zhi, Jin[3] ; & Xiaoguang, Yang [4]

(1, 4)Laboratory of Management Decision and Information Systems,
Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, 100190, China

(2) Department of Computer Science, University of Agriculture,
Abeokuta, Ogun state, Nigeria

(3)School of Electronics Engineering and Computer Science,
Peking University, Beijing, 100871, China

arogundadeot@amss.ac.cn[1]; aatakinwale@yahoo.com[2] ; zhijin@amss.ac.cn[3] and xgyang@iss.ac.cn[4]

Abstract

Abuse case has great support in identifying security threats and security requirements caused by outside attackers, but it has not been used to capture non-malicious deliberate acts for safety concerns that involves inside abusers. It is important to represent inside abusers in a model and distinguish them from inside intruders and outside attackers, since their behaviors are different. The intent of this paper is to propose a new extension of abuse case to identify deliberate acts of safety threats caused by inside abusers. A new notation vulnerable use case was introduced to express the actions that leads to threats from inside abusers, countermeasures were introduced by safety use cases, and new relationships were defined to clarify the interactions among use cases, vulnerable use cases, safety use cases and abuse cases. This enhanced model provided a way of capturing as much potential risks caused by inside abusers, and embed safety requirements in the early stage of the system development life cycle.

Keywords: *Abuse case, inside abuser, misuse case, safety threats, safety requirement, vulnerable use case*

1. Introduction

Information System (IS) safety has mainly been associated with unintended incidents (the accidental actions/mistakes). Previous work by different authors had actually addressed the accidental acts of safety but there are deliberate incidents in IS safety as well. In IS safety, the purpose of deliberate incidents is not directly related to the incident and consequences. The deliberate incident within IS safety do not want incidents to happen, but are made in order to do more work efficiently with less effort [1]. The deliberate acts can be characterized as cynical and ignorant (an example of this is the abuse cases initiated by the inside abuser which is our focus in this paper). Abuse case method provides support for the description of outside attackers [2]. But when an insider abuses his right, there might not be a malicious

intention to harm the system directly but it is a kind of risk that needed to be taken care of if the consequence is grave. This situation has not received a good description in literatures.

In this paper the ability of abuse case was enhanced to give a more complete overview of risk by adding some extensions, enabling the specification of safety threats by inside abusers. We applied this extension to E-health care system which is the motivation for the work presented in this paper. E-health is not an end in itself but a means to an end of achieving higher quality, safer, more value-driven and accessible health care for all. Thus in implementing E-health, as much risks as possible must be identified at the onset of the project.

The internet has transformed the patient-physician relationship by empowering patients with information. Physicians are no longer the primary gatekeepers of medical information. Since patients now possess more medical information, it is highly essential to ensure that patients do not abuse this privilege. This kind of privilege can promote harmful self diagnostic and treatment by patients. Also physicians can have the tendency of over reliance on Health Information System (HIS) which may make them to spend less time with their patients.

Designing a trustworthy solution to an environment like this, will entail capturing as much potential threats and abuses as possible. UML use cases are widely used technique for eliciting functional requirements when designing software systems [3]. The graphical style of UML furthers communication and interactions between different stakeholders involved in the development of Information System. Misuse cases have ways of revealing security, safety and non-functional requirements, as well as design tradeoffs to the surface. They illustrate the various relationships (<<threatens>>, <<mitigates>>, <<aggravates>>, <<exploits>> and <<conflicts with>>) between use and misuse cases allowing decisions to be made [4].

Abuse case was also proposed to elicit security threats and requirements [6]. This paper aims at enhancing the capabilities of abuse case for effective elicitation of safety threats and requirements by including inside abuser.

The contribution of this paper therefore can be measured in two ways; firstly we clarified that inside abusers are legal actors and capable of initiating vulnerable use cases though their intentions are not to harm the system but the consequences of their actions can be grave. Secondly, the paper enhanced the capability of abuse case model to identify deliberate acts of safety threats by inside abuser and capture safety requirements against these acts. The kind of safety threats considered here is not of accidental as stated by Sindre in [5] but that of accidental and deliberate without malicious intention.

The remainder of the paper is organized as follows. In Section 2, we give description of relevant terms used in the proposed model. In Section 3, overviews of related works were given for background study. Section 4 describes the proposed model and notations. It explains step-by-step approach on how to apply the extended notations and concepts, and also presents the use of the proposed notation in accordance to the aforementioned steps. We model safety concerns (deliberate and accidental) for E-health application using the proposed steps. We believe that this real world case study proves the capability of enhanced abuse case model presented in this paper. Section 5 gives the discussions and conclusions with the future areas of research.

2. Terminologies

This section identifies and presents basic constructs used in this paper and the proposed enhanced abuse case model. Also we try to differentiate between inside intruder and inside abuser. In the context of this paper we define the following concepts:

A use case is the specification of a set of actions performed by an actor [7]. This yields an observable result that is of value for one or more actors or stakeholders Use cases are the wanted actions and the actors are legal actors interacting with the system.

Abuse case can be defined as a kind of action which the system should not support. Abuse case is described from the point of view of an actor who wants to do work with less effort and seize some opportunities to benefit the organization but his intention is not to harm the system.

Vulnerable use case can be defined as use case whose initiation can lead to some abuse cases. Vulnerable use case prompts the unwanted actions of the inside abusers. We define a relation "aggravates" from vulnerable use case to abuse case.

Inside abusers are legal actors in the system. They initiate use cases which may eventually lead to some unwanted actions (abuse cases) but not with the intention of harming the system. Sometimes the consequences of these actions are grave.

The *safety use cases* for the system are the protective, detective and corrective actions mitigating the *abuse cases* in order to aid effective and efficient functioning of the system.

Rostad [8] in her work established the presence and actions of inside intruders. *She defined vulnerability as a weakness in the system which can be exploited. According to her an inside attacker/intruder is a legal actor that seizes the opportunity of vulnerability to carry out an illegal action with the intention to harm the system.*

There is a clear difference between inside intruder and inside abuser. The inside intruder has the intention to harm the system by his actions while the inside abuser does not have malicious intention. The inside intruder exploits a weakness in the system to carry out his malicious actions but the inside abuser ignorantly and cynically carries out his actions which may have grave consequences on the system. The inside intruder is not the legal actor of the use case that he exploited to carry out his actions but the inside abuser is the legal actor of the use cases which lead to the abuse cases.

The kind of chain actions between the inside abuser (who is also a legal actor) with the vulnerable use case helps in capturing safety threats and subsequent requirements. Understanding who the attackers are along with their motivations, goals, and targets aids designers in adopting proper countermeasures for the requirements in order to deal with the threats [9]. The details of these concepts are well reflected in the remaining part of this paper. In view of this we present the graphical view of use case, vulnerable use case, abuse case and safety use case in Figure 1.

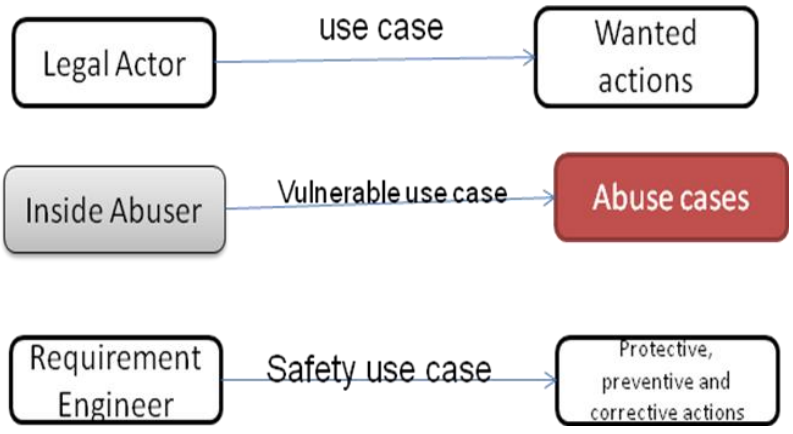

Figure1: Graphical representation of use case, vulnerable use case, abuse case and safety use case

3. Related Works

In the literature, security and safety are often taken to mean the same thing. In some stream of research the differences they claim between them are often very narrow to be comprehended. Some authors, e.g. Firesmith [10], used the term *security* for what concerns malicious (or deliberate) harms on IS, and used the term *safety* for what concerns accidental harms on IS. These authors used the broader notion of *survivability* to cover both security (in the above sense) and safety. The same notion was adopted by Sindre [5]. The notion of safety that we adopted in this work, which defines our scope, is broader. We thus looked at deliberate safety incidents distinctly. In the context of this paper we defined safety for what concern both accidental and deliberate incidents without malicious intention but may have grave consequences on the IS.

Use cases are among the widely accepted methods of eliciting, documenting and analyzing functionality requirements of systems. Equivalently, there are Misuse Cases and Abuse cases which are accepted as systematic means used for security threats and requirement elicitation.

Firesmith believed that "a misuse case is an effective method for analyzing security threats, but might not be well used for determining and analyzing security requirements". So he introduced *Security Use Case* and integrated it with UML diagrams [11]. There are other researches around Use and Misuse cases in [2, 12] that can be used to produce and model functional and security requirement. UMLsec [13], introduced by Jurjens, is a security improvement of UML. It models security features like confidentiality and access control. UMLsec includes all UML Analysis and Design artifacts like activity diagrams, deployment diagrams, sequence diagrams and state charts.

But the recent researches indicate that the main anxiety results not from the external but internal attacks and abuses [8].

Rostad also extended misuse case notation in eliciting security requirements by including vulnerabilities and insider threat [7]. In that paper, vulnerabilities are defined as a weakness that may be exploited by misuse cases. She pointed out the need to take care of the weak points in the system so that inside intruders will not be able to exploit them to affect the system in a negative way. The inside intruders are legal actors or stakeholders that carry out unauthorized actions in an illegitimate way to harm the system or for personal gain. She represented these vulnerabilities with grey color to differentiate them from the real misuse cases since they were opportunities created originally into system for effective functioning. She also introduced a new relationship <<exploits>> to link a threat to vulnerability.

In [6] Sindre presented a look at misuse case for safety concerns. He distinguishes security and safety based on the fact that security considers threats from malicious attackers while safety considers hazards resulting from accidental human or system failure. He used misuse case to identify accidental incidents for safety concerns and compare with other safety techniques. He established the relation 'aggravates' to link accidental acts for safety threats by the unlucky actor with misuse cases.

In this paper we investigate non-malicious deliberate acts for safety concern. In order to achieve this we introduce vulnerable use case and abuse case. The meaning of abuse case here is slightly different from the one describe by McDermott and Fox. The abuse cases here represent the deliberate acts for safety concern. These abuse cases are mostly orchestrated by the presence of administrative vulnerabilities. It is from non-malicious point of view.

McDermott and Fox [5] proposed 'abuse case'. Abuse cases are complementary to misuse cases. They focused specifically on security requirements and their relation to design. McDermott and Fox did not show 'use' and 'abuse case' in the same diagram so no

relationships between use case and abuse case could be depicted either. McDermott [14] defined abuse in terms of interactions that result in actual harm by an abusing actor. That is the abuser had the intention to harm the system from onset. He also commented that the actors in an abuse case model were the same kinds of agents that participate in use cases but they should not be the same actors. He only considered abuse case from malicious point of view.

Potts [15] distinguishes between 'abuse case' that violate policies and 'misuse case' that willfully undermine a policy e.g. using information for another purpose that it is gathered for.

3.1. Objectives

The objectives of this paper are as follows:

- To extensively exploit and enhance abuse case capabilities in capturing safety requirements.
- To distinguish between inside intruders and inside abusers.
- To enhance the capturing of as much potential risks caused by inside abusers.
- To identify inside abusive actions and specify countermeasures against them.

4. An Enhanced Abuse Case Model Including Inside Abuser

The originally proposed Abuse case by McDermott addresses outside attackers. This paper proposes an enhanced abuse case model and its notations including vulnerable use case and inside abuser.

Figure 2 illustrates the conceptual view of the proposed model while figure 3 presents the overview of the abuse case notations, the actors involved and the relationship they exhibit between one another.

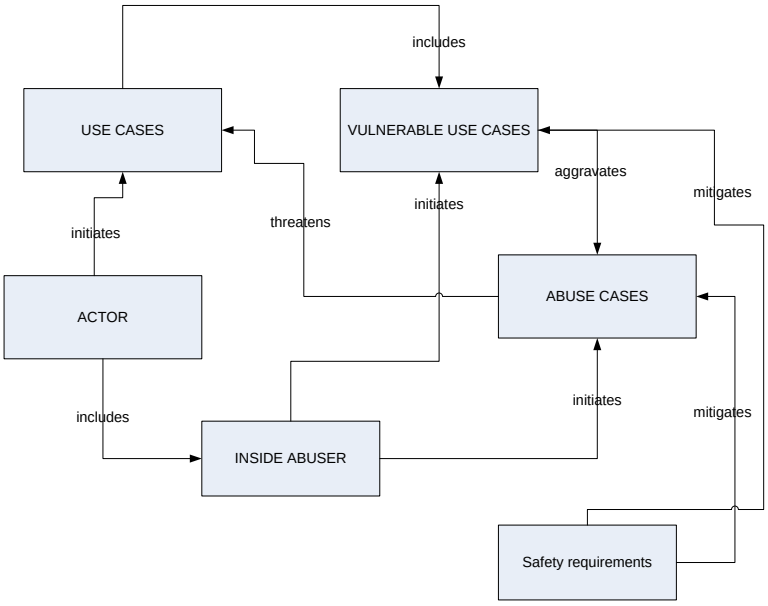

Figure 2: Conceptual View of the Proposed Model

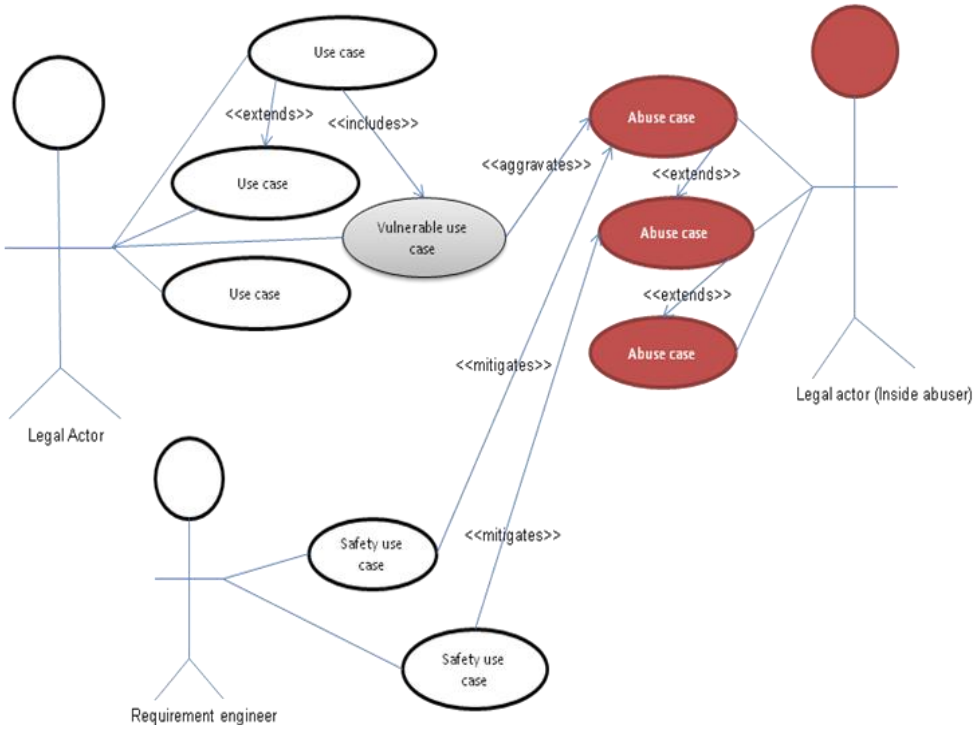

Figure 3：The enhanced Abuse case notations including inside abuser for safety concerns

In this figure, the authorized actor is the white stick man on the left hand side. The white oval shape notation describes what the actor can do with the system (use case). The use case exhibits two relationships <<includes> and <<extends>>. <<includes>> is the relation that exist between a use case and vulnerable use case and << extend>> is a kind of specialization relations that exists between one use case and another of the same actor. The grey oval shape notation is the identified vulnerable use case, whose initiation by the actor can lead to abuse cases. These abuse cases are the red oval shape notations. There are three different relations associated with the abuse case notation. They are: <<extends>>, <<threatens>> and <<aggravates>>. <<Extends>> is a kind of specialization relation between an abuse case and another abuse case. As seen in real world case illustrated in this paper in figure 4, an abuse case may lead to another abuse case. The aggravate relation exists between the vulnerable use case and the abuse case, while abuse case threatens the use case. The oval shape notation shown at the bottom towards the left hand side of the diagram is the safety use case. It exhibits relation <<mitigates>>. It mitigates the abuse cases. The red stick man on the right hand side is the inside abuser who is also the legal actor in the system as shown in the diagram.

4.1. Enhanced Abuse Case Modeling Processes

In this section we present the overview of the modeling process of the proposed model. This is shown in figure 4.

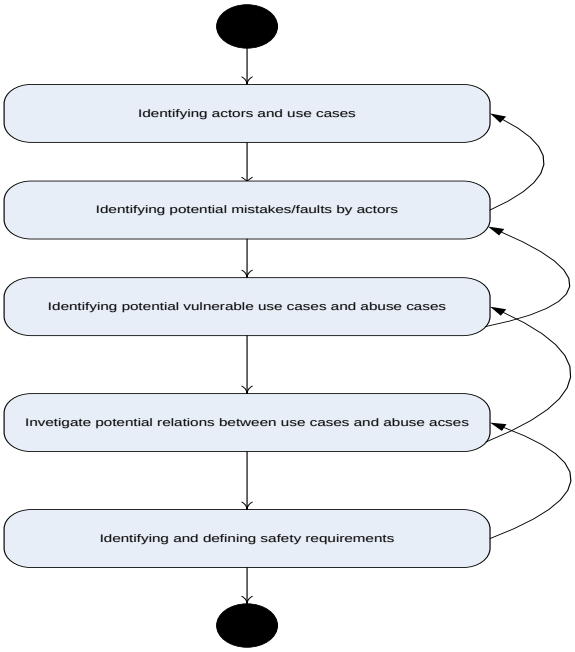

Figure 4: Overview of the Modeling Processes

The steps involved are elaborated below.

Step 1: Identify the normal actors and their interactions with the system that is the use cases and specify safety goals for them.

Step 2: Identify possible mistakes that may be committed by the actors in the course of interaction with the system. This will take care of accidental incidents of safety threats.

Step 3: Identify possible vulnerable use cases for each of the actors. This can be achieved by examining each of the use cases for the possibility of being abused by the legal actors. These vulnerable use cases motivates the unwanted actions (abuse cases) of the inside abusers. These abuse cases are deliberate incidents of safety threats which are our major concern in this paper.

Step 4: Investigate the potential relations between abuse cases and use cases. This is very important since many threats to a system can largely be achieved by using that system's normal functionality as for instance "harmful self treatment" abuse case of our example.

Step 5: Identify and define safety requirements for the risks posed to the information system by the abuse cases. Safety requirements can be identified by considering the assets affected and the safety goals of the assets. We suggest that safety requirements against deliberate acts of safety can be captured by considering both functional requirements and the business rules or goals guiding the information system. as a result of this the context within which the system will be put into use is very important. To illustrate consider some software used by a medical director on her desktop computer. The software might not have intrinsic safety goals associated with it, but the information the medical director manipulates may have to be error free creating a maintain integrity safety goal for the system comprising the medical director and more importantly the patients that has the data. The safety goal arises because of the context within which the system is being used. In order to satisfy this goal from functional requirement point of view one might consider mandating the medical director to double check

every input data for correctness or every input is sent to a second party for checking. Alternatively, one might decide that the software should satisfy the goals perhaps by adding automatic spell checking for all kinds of drugs being administered or that exist. However these solutions are inadequate if the medical director is less skilled in the use of the software or does not have enough training on safety awareness. The captured safety requirements are represented as safety use cases in the model. The example we use to illustrate our proposed model in section 4.2 expresses identifying safety requirements from the point of view of business rule/goal.

A near optimal solution is given by repeating Steps 1 through 5 for detailed and complete investigation.

4.2. Illustrative Example (e-healthcare system)

This section presents a small part of the example. The e-health care information used in this paper is retrieved from Health Information Technology Resource Toolkit developed through the university of Kansas HISPC project. http://ehealth.kansashealthonline.org. We use the Health Information System (HIS) from this project as introductory example and as a validation of our work.

The HIS works as follows. HIS allows patients to have sole right and access to their medical records. Physicians also have access to all information they need in order to provide medical treatment for patients. From the patients point of view there are principal use cases: (1) request for emergency (2) access medical record (3) receives quality medical treatment (4) renew medication. From the physician point of view we have the following use cases (1) prescribe drugs (2) attend to emergency (3) provide medical treatment (4) access patient medical records.

A simple example of deliberate acts for safety concerns would be if patients go into dangerous self diagnostic and self treatment which may lead to death. Self diagnostic is against the health care information business rule. This may be possible since the patients have sole right and access to their medical records. Another example would be physicians over reliance on the health information system and then fail to visit the patients regularly or even spend less time with them if they visit at all. This may lead to patient under treatment in the case of a patient just developing a new symptom of a disease which has not been investigated or recorded.

We show in the remainder of this section how these deliberate acts and possible countermeasure can be modeled using the enhanced abuse case model proposed in this paper according to the steps highlighted in section 5.

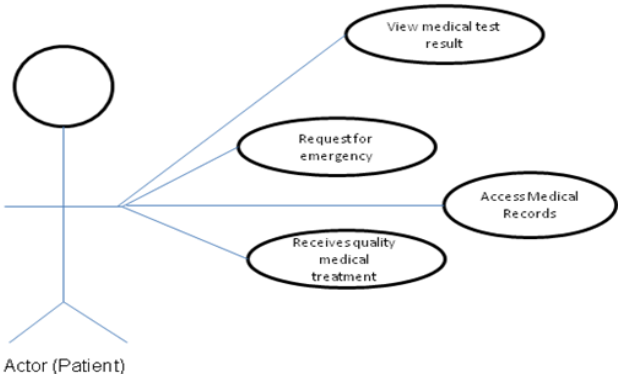

Figure 5(a): Patient (actor) and the Use Cases

Figure 5(a) shows the actor (Patient) and the use cases (request for emergency, access medical records, receives quality medical treatment and view medical test result).

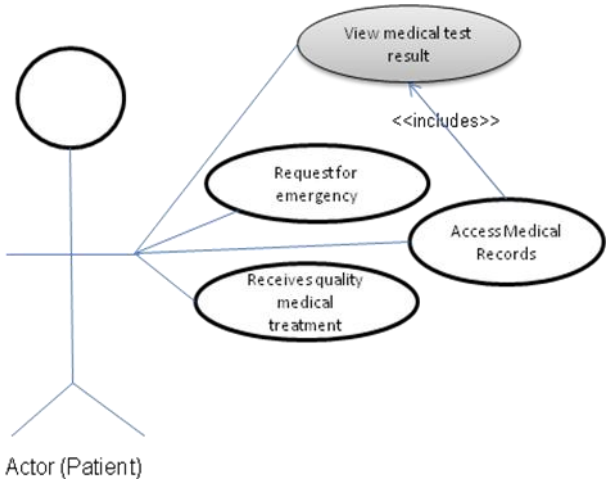

Figure 5(b): Patient (actor) and the identified vulnerable use case

In figure 5(b) the use case "View medical test result" is identified to be a vulnerable use case because it can lead to some unwanted actions by the actor (patient). The possibility of this unwanted actions are explained in figure 5(c). The relation between use case and vulnerable use case is also shown.

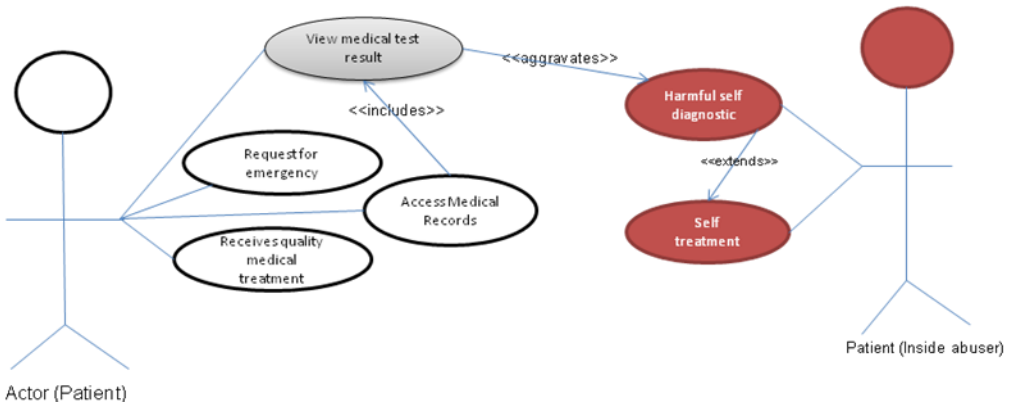

Figure 5(c): Patient (actor) initiating an abuse case as a result of the vulnerable use case

Figure 5(c) shows how the actor (patient) initiated abuse cases "harmful self diagnostic" and "self treatment" as a result of the vulnerable use case in figure 5(b). Since the same actor in the system is initiating a abuse case as a result of vulnerable use case which he once initiated, then he is referred to as an "inside abuser".

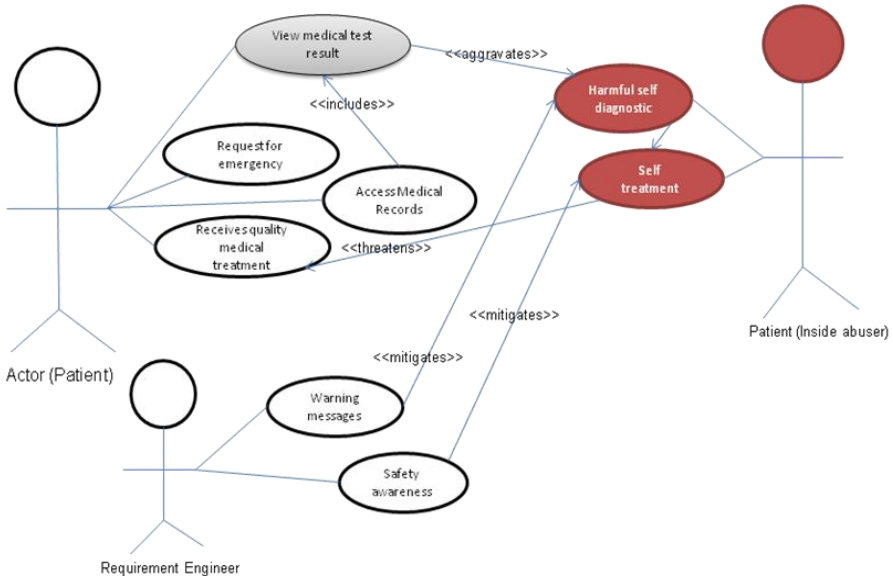

Figure 5(d): Abuse case relation to use case and safety use case

Figure 5(d) shows how the abuse cases relate to the receive quality medical treatment use case. We use standard misuse case relationship. An abuse case <<threatens>> a use case if it potentially could prevent the use case's goal from being achieved. A mitigating case (that is safety use case) <<mitigates>> an abuse case if it reduces the possibility of the unwanted action being successful.

If the abuse cases in figure 5(c) are carried out successfully, then the trend of medical treatment for the patient cannot be monitored or followed through. In this wise, use case 'receive quality medical treatment' is threatened. In this figure also two safety requirements were defined to mitigate the abuse case they are: warning messages and safety awareness. In the original definition of the use cases for the patient he never receives warning messages on safety awareness and medical ethics. By introducing safety awareness and warning messages the possibility of carrying out the abuse cases can be reduced.

We shall consider another example in which the Physician is the actor.

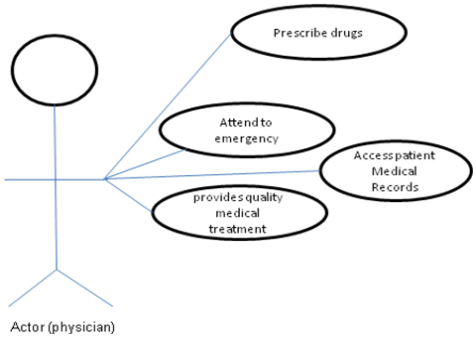

Figure 6(a): Physician (actor) and some of the use cases

In figure 6(a) the actor is the physician and in his interaction with the health information system he carries out the following actions which are parts of the use cases (1) prescribe drugs (2) attend to emergency' (3) provides medical treatment (4) access patient medical record.

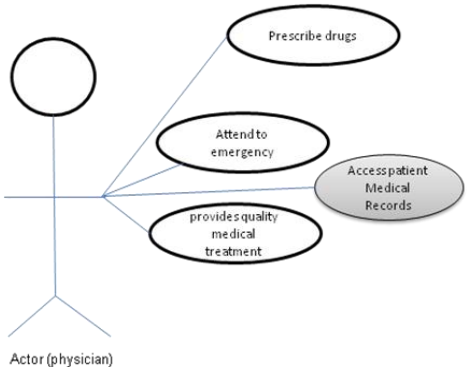

Figure 6(b): Physician (actor) and the identified vulnerable use case

In figure 6 (b) the use case access patient medical record is identified to be a vulnerable use case because it can lead to some unwanted actions (abuse case) by the physician himself.

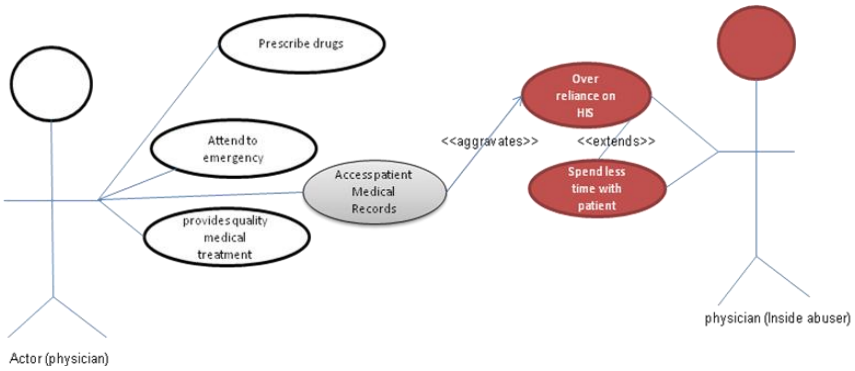

Figure 6(c): Physician (actor) initiating abuse cases as a result of the vulnerable use case

Figure 6(c) shows the physician initiating abuse case 'over reliance on HIS' because of the vulnerable use case indentified in figure 6(b). This abuse case extends to misuse case 'spend less time with patient' and invariably lead to misuse case 'patient under treatment'.

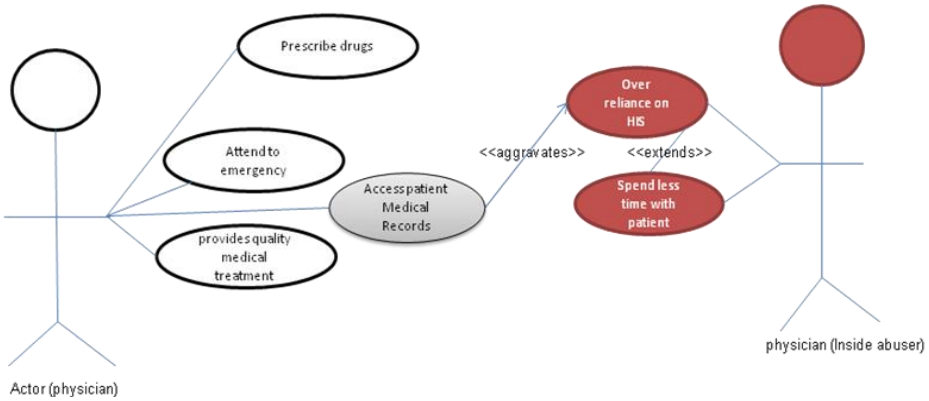

Figure 6(d): Misuse case relation to use case and safety use case

In figure 6(d) the abuse case spend less time with patient is a threat to the use case 'provides quality medical treatment'. These abuse cases are initiated by the physician himself who was the legal actor but also the inside abuser. We define safety tools 'safety awareness' and 'performance evaluation' as safety requirements against these abuse cases and vulnerable use case. The abuse cases initiated by inside abuser as a result of vulnerable use case helps in eliciting safety requirements. We discovered that the consequences of successful initiation of these abuse cases are related to human injuries/death. This agrees with the claim of Molman in his article [1] that loss as a result of safety threats is related to human injuries/death and reliability of organizational assets. We also discovered that the relation 'aggravates' best describe the relationship that exist between the vulnerable use cases and abuse cases initiated by the inside abuser. This comply with the work of Sindre in [5] where the relation 'aggravates' links the accidental acts of safety by the actor with the misuse cases.

Though most patients and physicians may not attempt to commit these errors, but it is not impossibility. It is therefore necessary to consider the possibility of these and map out potential consequences and countermeasures if the consequences are grave. This is why enhancing abuse case model with vulnerable use case introduced by inside abusers in order to elicit safety requirement is very essential. By identifying the possibility of inside abusers we are able to include proper countermeasures during system development.

5. Discussions

The idea expressed in this article, came up during a research project on E-Health care system that has to do with investigating and identifying security and safety requirements in E-Health care system.

The originally proposed abuse case by McDermott has much and greater support for outside attackers. In addition to Sindre work this article has extended and enhanced the capabilities of abuse case to include inside abusers in order to capture deliberate acts of safety threats. What is unique in this work is that we have been able to prove that inside abusers can be legal users or the same actor of the use case in the system. To the best of our knowledge, this has not been described in literature. Rostad work focuses on security requirements including vulnerabilities but in this case we have been able to discover that inside abusers actions help to elicit safety requirements which are different from security requirements with the inclusion of the vulnerable use case. Table 1 gives a brief comparative analysis of the existing models with the current proposal. In table 1, we use'**+**' to indicate the explicit support for each of the concepts described in the first row and '**¬**' to indicate partial support while '—' indicates no support. The purpose of this comparison is not to determine which of the extensions is best but to distinctly present the focus and the kind of threats each addressed.

Table 1 Comparison of some existing models with the current proposal.

Author	Relation of vulnerability to use case	Vulnerability analysis approach to security	Vulnerability analysis approach to safety	Accidental acts of safety threat	Deliberate acts of safety threat
Rostad, 2006	+	+	–	–	–
Sindre, 2007	–	–	¬	+	–
Current Proposal	+	–	+	+	+

6. Conclusions and Future Work

In this article we have been able to show the relationship and interactions between ‘use ’, ‘vulnerable use’ and ‘abuse’ cases in the same diagram for a better understanding, and elicitation of safety threats and requirements. In [6] abusive action was viewed from a malicious point of view, but in this paper we have been able to identify some abusive actions (e.g. self treatment, over reliance on HIS) whose intentions are not to harm the system directly but pose a kind of risk to the system stakeholders because the consequences are grave.

Enhancing abuse case model to include inside abusers can be very interesting and essential. It allows for capturing more potential risks in the system. The complemented notation enhances the capabilities of abuse cases to give more support for inside abusers and their actions which can be harmful to the system and the stakeholders. Though the act of abusing legal right is not a direct attack on the system but they are risks that must be taken care of in the process of system development. Hence providing a better concept of the extent of risks and how to handle them are necessary. The model proposed in this paper is very simple and easy to understand. The reasoning behind each of the requirements can immediately be comprehended by consumers because of the visual description of the interactions among the actors, use cases, vulnerable use cases and abuse cases.

By identifying safety threats and requirements early in the development process and incorporating them throughout the system development life cycle, a trustworthy system will easily be released. Also and most importantly the system will benefit from stronger security and safety, reducing the likelihood of abused vulnerable use cases. Moreover, a properly constructed abuse case model can help architects identify architectural changes in the design phase, help developers understand the non-malicious user's approach to write more secure code.

For the further area of research, we shall incorporate this extended and enhanced model into the original model of misuse case in order to capture security and safety requirements in the same conceptual framework. Furthermore, we aim to give a formal representation of the knowledge of the proposed model to allow for automation of the activities involved in the modeling process.

Acknowledgements

The research is supported by the Key Project of National Natural Science Foundation of China under Grant No.90818026, Swedish International Development Cooperation Agency (SIDA) and Organization for Women in Science for the Developing World (OWSDW).

References

[1]. Mølmann, R.A., “The Human Factor – Taxonomy for classifying human challenges to information security”. In Kufås, I and Mølmann, R.A., Informasjonssikkerhet og innsideproblematikk. Institutt for produksjons- og kvalitetsteknikk, NTNU, 2003.

[2]. I. Alexander “Misuse cases, use cases with hostile intent”, IEEE software 20(1): 2003, pp. 58-6.

[3]. Sindre G. and A.L.Opdahl “Eliciting security Requirements by misuse cases”. Requirements Engineering 10 (1): 2005, pp. 34-44.

[4]. I. Alexander, “Misuse Cases Help to Elicit Non Functional Requirements”, Computing & Control Engineering Journal, vol. 14(1), Feb. 2003, pp. 40-45.

[5]. Sindre, G., “ A Look at Misuse case for Safety concerns” in IF[P International Federation for Information Processing, Volume 244, Situational Method Engineering: Fundamentals and Experiences, eds. Ra[yt~, .i., Brinkkemper, S., Henderson-Sellers B., (Boston Springer), 2007, pp. 252-266.

[6]. McDermott J, Fox C. "Using abuse case model for security requirements analysis". In Proc. of the 15th Annual Computer Security Applications Conference (ACSAC '99), Phoenix, Arizona, 1999.

[7]. I. Alexander and N. Maiden "Scenarios, stories, use cases: Through the systems Development Life-cycle". John Wiley and Sons, 2004. ISBN: 0470861940.

[8]. Lillian Rostad "An extended misuse case notation: including vulnerabilities and the insider threats". In Proc. Of REFSQ, Luxembourg, 2006.

[9]. B. Schneier. Attack trees. Dr. Dobb's Journal 24(12): 1999, pp.21-29.

[10]. D. G. Firesmith, "Common Concepts Underlying Safety, Security, and Survivability Engineering," Technical Note CMU/SEI 2003TN-033, Software Engineering Institute, Pittsburgh, Pennsylvania, December 2003.

[11]. Donald G. Firesmith, "Security Use Cases", Journal of Object Technology, vol. 2, no 3, May-June 2003, pp 53-64, also available at http://www.jot.fm/issues/issue_2003_05/column6

[12]. I. Alexander. Modeling the interplay of conflicting goals with use and misuse cases. In Goal-Oriented Business Process Modeling (GBMP), CEUR Workshop Proceedings Volume 109, London, UK, 2002.

[13]. J. Jurjens, "Towards Secure Systems Development with UMLsec", Fundamental Approaches to Software Engineering (FASE/ETAPS) 2001, International Conference, Genoa 4-6 April 2001.

[14]. McDermott J. "Abuse-case-based assurance argument", In proc. Of the 17th annual computer security applications conference (ACSAC'01), New Orleans, Los Angeles, 2001.

[15]. Potts C. Scenario Noir (Panel statement, p2) In Proc. of the symposium on requirements engineering for information security (SREIS '01), Indianapolis, 2001.

Authors

Arogundade Oluwasefunmi is a PhD candidate at Academy of Mathematics and System Sciences, Graduate University of Chinese Academy of Sciences, Beijing China. She received a B.Sc. degree in computer science from the university of Ado-Ekiti, Ekiti State, Nigeria. She had the M.Sc. degree in computer science from the University of Agriculture, Abeokuta, Nigeria. Her current research interests include requirement engineering, reuse, ontology, business /IT alignment and information management science. She had published many articles in journals and conferences.

Adio Akinwale works as an associate professor and researcher at University of Agriculture, Abeokuta Nigeria. His research interest encompasses Management Information System, Query Algorithms Optimization and cybernetics. Adio holds Master degree in Cybernetic and Computer Science and Ph.D. degree in Economic Informatics all from Oskar Langer University, Wroclaw, Poland. He had published many articles in journals and conference proceedings.

Zhi Jin received the MS degree in computer science and the PhD degree from the Changsha Institute of Technology, China, in 1987 and 1992 respectively. She is now a professor of computer science in the Academy of Mathematics and System Science, Chinese Academy of Science. Her current research is on software requirements engineering and Knowledge Engineering. She has published more than 90 referred papers in the area of requirements engineering and knowledge-based software engineering.

Xiaoguang Yang is a full professor at Institute of System Sciences, Academy of Mathematics and System Sciences, Chinese Academy of Sciences (CAS). He is currently the director of key laboratory of management Decision and Information system (MADIS), CAS. Prof. Yang research interests include risk management, operation research, and information system. He had published many papers in both domestic and international journals. He has long years of working experience both as an academics and practitioner.