OPEN ACCESS 1

ISSN 2460-9056
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 7, Issue. 01, June 2021. pp. 1-9
doi: 10.34818/ijoict.v7i1.543

Application of the Misuse Cases Method to Analyze Safety Gaps in an Electric Car Auto Drive Case Study

Ryo A. Ramadhan [1], Dana S. Kusumo [2], Jati H. Husen [3*]

[1,2,3] *School of Computing, Telkom University*
Bandung, West Java, Indonesia

[*] jatihusen@telkomuniversity.ac.id

Abstract

Safety requirements analysis is an activity inside software requirements engineering that focuses on finding and solving safety gaps inside a software product. One method to do safety requirements analysis is misuse cases, a technique adopted from the security analysis method. Misuse cases provide a safety analysis approach which allows detailed steps from different stakeholders' perspective by showing a series of unwanted actions during usage of a software product from different users' points of view. This research evaluates the misuse cases method's understandability by implementing it to analyze safety requirements for an electric car's autopilot system. We assessed the developed models using the walkthrough method. We found differences between how the model understood by someone with experience in software development and those who don't. This gap is likely caused by how the software development experience allows the model's user to interpret the model effectively and understanding the detailed information on how each risk works. Improving the model by adding more mitigation cases based on user point of view to minimize adverse effects from mistakes and errors also eliminating unnecessary detail and ambiguous notation is expected to make it easier for users to understand the misuse cases model.

Keywords: *Misuse cases, Requirements engineering, Safety requirements analysis, Use cases.*

I. INTRODUCTION

Safety is a critical quality attribute of software. The importance of safety leads to the rise of software safety requirements, which serve as a basis for developed software products [1]. Safety requirements are a defined basis that aims to minimize risk by making faults prediction, which may happen before launching the product [2]. A reasonable software safety requirement can be shown by how much the user accepts the different cases of identified cases [3].

One way to identify safety requirements is to analyze safety characteristics described in existing documentation [4]. The analysis is conducted by several aspects, including functional, system failures, impact analysis, and preventive action for each fault and mistake [5]. Several methods are currently introduced to analyze software's safety, such as fault trees, HazOp, and FMEA, with each method having different strengths and weaknesses [6]. However, those methods do not highlight software safety from the user's point of view, which is essential in designing preventive actions [6].

Received on 8 Jan 2021, Revised at 13 Mar 2021, Accepted at 19 Mar 2021, Publish online on 18 June 2021

The misuse cases method for the safety gap analysis was chosen based on Ophdal's research on how misuse cases use the user perspective to analyze security gaps by displaying unwanted actions that can occur [7]. Misuse case is a safety analysis method extended from the use case model that shows a series of unwanted activities during usage of a software product from different users' points of view [8]. Formerly used for analyzing security threats, misuse cases have been adopted to identify internal threats from user mistakes and system failures [7]. Implementation of misuse cases itself contains several steps, from breaking down existing requirements, developing UML diagrams, analyzing potential failures, and finally validation to finalize the developed misuse case model [6].

This research will investigate the capabilities of misuse cases in modeling software safety requirements. We will implement the misuse case method to analyze the safety of an electric car's autopilot software system. The case selection is based on data that shows the critical risk of driver safety when the autopilot system fails in real-life scenarios [9, 10]. This research will focus on developing a misuse cases model for the autopilot module based on user error protection and fault tolerance defined in ISO 25010 [11]. Developed misuse case model then evaluated by walkthrough with users [12]. Inputs from users then will be discussed to find the gap between safety requirements shown in the developed model and user's expectations.

To achieve the goal, we devised the following research questions to guide this research:

- RQ1. How good is misuse case, both diagrammatically or textually, in showing potential failure and user mistakes?
- RQ2. What kind of improvement can be done to improve the misuse case method?

Our research contributes to software requirements engineering areas in the form of knowledge of benefits from misuse cases for safety requirements analysis.

The rest of this article is constructed as follows: section 2 describes works and theories related to this research. Section 3 explains the activities throughout this research process. Section 4 presents the results of this research and discusses them. Finally, section 5 concludes this article and proposes future directions for research in misuse cases method.

II. LITERATURE REVIEW

A. Software Safety

The concept of safety is prominent in software engineering. Zielenski describes the basic concept of safety, such as the definition of safety itself, safety requirements, and safety constraints [13]. Zielenski also explains what kind of quality factor and subfactor needs to be attended to, including metrics and criteria that can be used to analyze a product's safety. Donald explained that safety consists of 3 main factors that become the basis of a product design process [14]. Those factors are health, property, and the environment.

Several methods can be utilized to analyze software safety factors, such as fault trees, FMEA, and HazOp. Each method has a different approach, which provides various benefits and challenges. Fault trees follow a top-down process in its implementation, focusing on the cause of the problem to conduct the recovery process [15]. Each step run in fault trees is done to identify issues that developed software will faces, grouped into possible and certain problems. HazOp (Hazards and operability analysis) utilized textual representation to improve the analysis's relevancy [6]. HazOp divides a system into smaller steps, in which every step will be tested for possibilities of mistakes and errors [16].

B. *Safety Analysis Method*

The idea of adopting a misuse case for safety analysis is proposed by Guttorm [17]. His paper demonstrated that the implementation of misuse cases also shows safety problems, not just security problems. Adoption is achieved by exploring threats from failures of the system itself and mistakes made by users. Arogundade demonstrated enhancement to misuse case method by introducing iterative risk assessment to improve proposed preventive actions [1].

One of the main advantages of using misuse cases for security is to show a series of unwanted actions from different perspectives sequentially [1]. By adopting misuse cases, the same benefit is gained for safety analysis. This research is conducted to prove the claim further.

III. RESEARCH METHOD

Figure 1 shows an overview of our research method. Our research adopts the first three phases of Arogundade's study [1] and adds an elicitation phase in the beginning. Elicitation is conducted by interviewing users to elicit information about existing behaviors and possible mitigations for each risk.

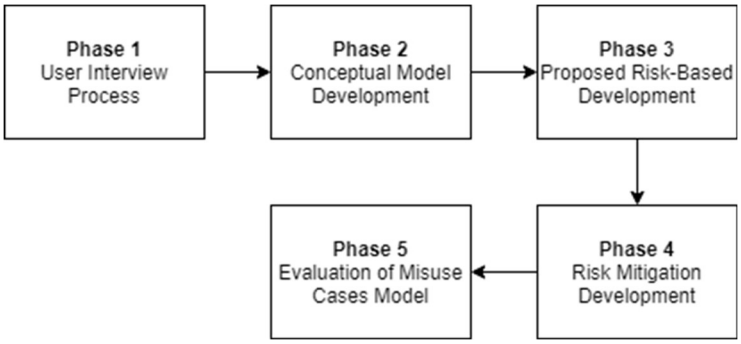

Fig. 1. Research Phase

A. *User Interview Process*

User interview is conducted to collect actual data from the users' point of view. The interview is conducted with five users with different backgrounds to allows variation of data and minimize bias towards any interview questions. As per general rules, our target user is someone aged between 20-50 years old, holding a car driving license, and experienced in driving for at least four years. Table I shows the interviewee of the interview activity.

TABLE I
STAKEHOLDER PROSES INTERVIEW

ID	Age (year)	Driving Experience (year)
Source1	48	15
Source2	21	5
Source3	22	7
Source4	22	5
Source5	22	6

The interview questions are designed around driving behavior, steps and process of driving, and risk mitigation for each driving case. Driving cases in our research are based on seven driving maneuvers stated in Caroline's research on autonomous driving behavior [18]. The case also further scoped into lane-changing maneuver in city road conditions [19]. Table II shows the sample of questions asked in the interview for each driving case. Follow up then conducted based on the results of the interview. The follow-up is performed to resolve the differences between different users for each case.

TABLE II
DRIVING OPERATION AND MANEUVER

Driving Case	Interview Question
Follow the Road.	What is the sequence of action when you follow the road?
Lane Changing and Overtake Other Vehicles.	What are the potential problems that may occur when changing lanes and overtaking other vehicles?
Turn Left/Right.	Based on previously stated problems, how often those problems occurred?
Crossroads.	What is the sequence of action you do when facing a crossroad?
Obstacles.	What series of actions did you do when facing an obstacle?
Traffic Jam Conditions.	What kind of problem do you face when stuck in a traffic jam?
Reverse and Parking.	What kind of risks exists when failing to execute reverse or parking properly?

B. *Model Development*

We developed the misuse case model in three phases. The first phase is the development of conceptual use case model based on the results of the interview. After that, we develop a proposed risk-based use case model that shows potential failures and mistakes. Finally, we finish the misuse case model by adding risk mitigation for each potential failure and error. Figure 2 explains the notations used in our misuse case models [8], [20].

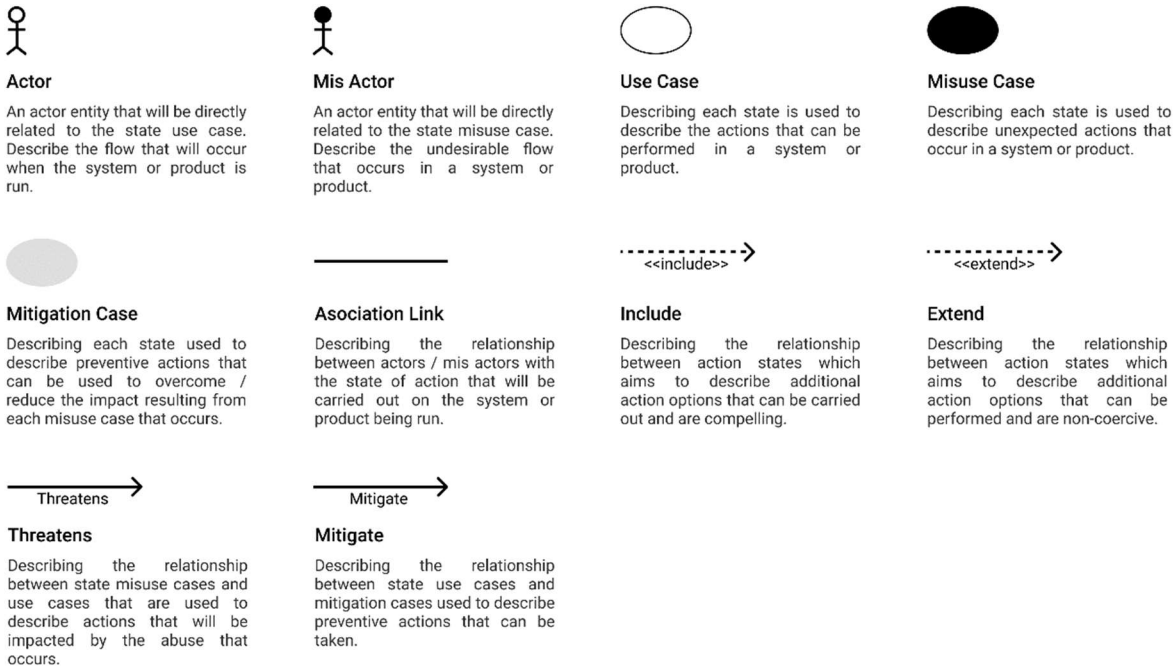

Fig. 2. Misuse Cases Component Details

C. *Evaluation of Misuse Cases Model*

Evaluation is conducted via a walkthrough requirement validation technique to find mistakes inside the model [13]. The validation is performed with reviewers from both general users and expert representatives. Both general users and expert representatives are familiar with UML models. However, experts are selected by their experience in developing software. The difference between the two groups will indicate whether the model's acceptance is varied by experience. Table III describes each participant's background; each participant's name is obscured to protect their privacy.

TABLE III
WALKTHROUGH PARTICIPANTS

Role	Background
Author/Reader	Bachelor Students
Notetaker	Bachelor Students
Expert1	Quality Assurance Engineer at WHIM Management
Expert2	Back-end Developer at ChatAja
Expert3	IT Manager at LawyerIndo
User1	Bachelor Students
User2	Bachelor Students
User3	Bachelor Students

The evaluation process is divided into 2 phases. The first phase follows a typical walkthrough process, explaining the misuse case model with experts and users giving comments and questions about several aspects of the model. All responses will then be discussed further in the second phase to find the causes behind those responses. The result of the discussion will then be analyzed to measure the capability of the misuse case.

IV. RESULTS AND DISCUSSION

A. *Developed Model*

TABLE IV
DESCRIPTIVE MISUSE CASES SAMPLE

Name	Follow The Road Case
ID	1
Description	The auto-drive system directs the vehicle to follow the path based on the direction entered and takes into the surrounding conditions
Actors	Auto drive system
Triggers	The direction of the destination and surroundings
Pre-Condition	The driver enters the driving direction
Post Conditions	The vehicle will go to the destination based on the direction entered and the surrounding conditions
Basic Flow	- The system synchronizes destinations and paths used - The system detects the state of the road and surrounding vehicles - Conditions of roads and surrounding vehicles are safe - The system regulates vehicle speed
Alternate Flow	- The system synchronizes destinations and paths used - The system detects the state of the road and surrounding vehicles - There are obstacles in the condition of the roads and surrounding vehicles - The system regulates vehicle speed - The system prepares options for countermeasures

The misuse case descriptive model is designed based on the evaluation results obtained at the user interview process. The descriptive model that is designed contains information about the action taken, the actors involved, the triggers that trigger the action, and several other components that can explain the flow of the action taken in the existing case. Table 4 is a sample of the designed descriptive model of misuse cases. A complete model regarding the developed misuse cases can be seen in the research data resource in section 6.

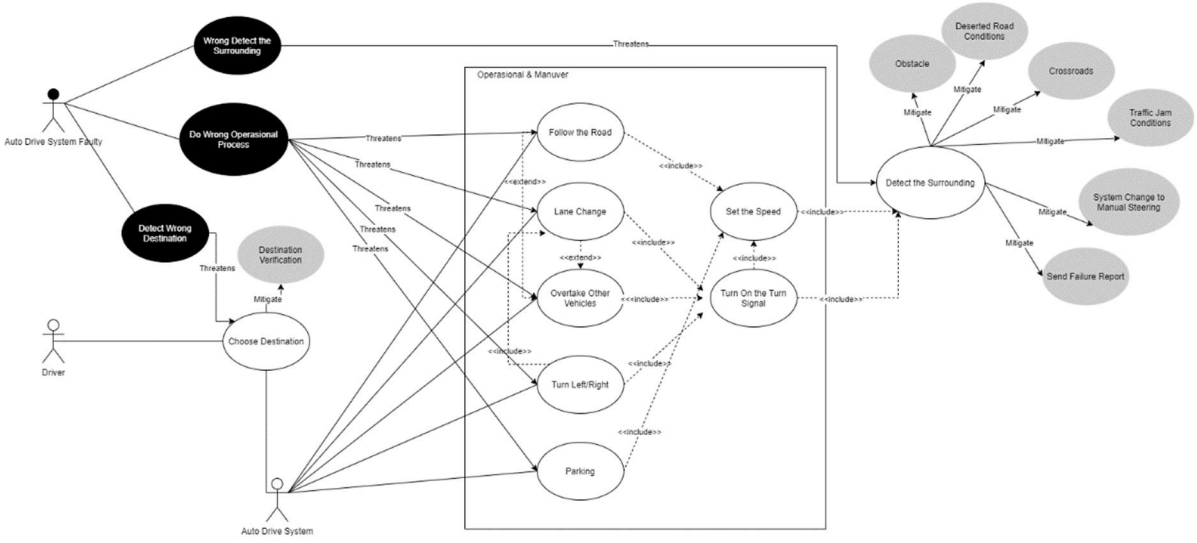

Fig. 3. Overall Misuse Case Model

Figure 3 shows the developed misuse case model for this research. There are two actors identified, the first one is the driver itself and the second one is the auto-drive system installed inside the autonomous car system. The identification of actors is based on safety analysis limitations, where there are only two relevant entities: users' making a mistake and failure of the system [6]. Conceptual use cases and their association with actors are devised from 7 driving maneuver described in Table II and user interview results. Misuse cases are originated from interview results about potential problems that may occur from drivers' mistakes and system failures. Finally, mitigations are implemented based on the answers to interview questions about risk mitigation commonly applied.

B. *Evaluation Result*

Table IV summarizes the evaluation result. Each response from participants is classified into positive (+) and negative (-) results. The detail of responses for each participant is available in our research data repository presented in section 6.

TABLE V
EVALUATION RESULTS

No	Category	Expert 1	Expert 2	Expert 3	User 1	User 2	User 3
1	Understandability of detailed steps from developed misuse cases.	+	+	+	+	-	-
2	Understandability of sets of problems from developed misuse cases.	+	+	+	+	+	+
3	Understandability of series of mitigations from developed misuse cases.	+	+	+	+	+	+
4	The capability of the developed model to be improved.	+	+	-	+	-	-

From the experts' point of view, there is no problem with the developed misuse case model. All expert representatives are showing positive reactions to the understandability of the developed model. There are only two problems during the walkthrough: the unfamiliarity of the notation used and the model's overall size. Two experts agree that a developed model can be improved with more mitigation cases to minimize adverse effects from mistakes and errors.

For general users, the results are less favorable. While they generally understand the model, two representatives have difficulties understanding the detailed steps shown in the misuse case model. The problem came from a lack of information on how each misuse case is decided. When combined with a lack of expertise in software development, those gaps impeded the understandability. The only general user who understands the model suggested adding another detailed information on the developed model to tackle that problem.

C. *Discussion*

For the answer to RQ1, we found differences in how understandable the model for general users' and experts' perspectives. The results are overwhelmingly positive from the experts' perspective, which indicates that the misuse case is highly usable for someone with software development experience on top of familiarity with UML models. The only challenge is that an introductory explanation is needed for misuse cases to be implemented for this group of users. On the other side, general users require additional materials to be able to use the model effectively. This gap is likely caused by how the software development experience allows the model's user to interpret the model effectively. Based on the researcher's point of view, since the issue is in the comprehensibility of the model, adding more documentation/model may make it worse. One must think to reduce complexity by eliminating unnecessary detail, ambiguous notation, and other challenges.

For the answer of RQ2, misuse cases may require additional models or documentation for more effective usage. Use of another UML model like activity or state diagram and flowchart may improve the misuse case model's understandability. This suggestion comes from how general users asked for more explanation on detailed steps for each risk. The other improvement is to introduce other risk analysis methods to find more options for minimizing risks. This improvement is based on experts' comments on how risk mitigation can be increased.

V. CONCLUSION

We conclude that misuse cases can be utilized as a software safety requirements analysis method with some limitations through the evaluation. This conclusion is based on our finding that the misuse cases method is highly effective for users with software development experiences. However, it is not as effective when the users lack software development experiences, especially in understanding the detailed information on how each risk works.

In the future, research on how to improve the model by adding more mitigation cases based on user point of view to minimize adverse effects from mistakes and errors. We will also aim to eliminate unnecessary detail and ambiguous notation to make it easier for users to understand the misuse cases model. Another direction is to find how different risk analysis techniques can improve the risk identification and mitigation decision-making inside the misuse cases method.

VI. DATA AND COMPUTER PROGRAM AVAILABILITY

Interview result data is available at http://gg.gg/MisuseCasesInterviewResultData. The Misuse case model for each case can be accessed at http://gg.gg/MisuseCasesModel. Evaluation data were available at http://gg.gg/MisuseCasesEvaluationResultData.

REFERENCES

[1] O. T. Arogundade, S. Misra, O. O. Abayomi-Alli, and L. Fernandez-Sanz, "Enhancing Misuse Cases with Risk Assessment for Safety Requirements," *IEEE Access*, vol. 8, pp. 12001–12014, 2020, doi: 10.1109/ACCESS.2019.2963673.

[2] O. M. Kirovskii, and V. A. Gorelov. "Driver assistance systems: Analysis, tests and the safety case," ISO 26262 and ISO PAS 21448. *IOP Conference Series: Materials Science and Engineering*, vol. 534*,* no. 1, 2019, doi: https://doi.org/10.1088/1757-899X/534/1/012019

[3] I. Alexander, "Misuse cases help to elicit non-functional requirements," *Computing and Control Engineering Journal*, vol. 14, no. 1, pp. 40–45, 2003, doi: 10.1049/cce:20030108.

[4] Y. Wang, *et al*, "Research on Formal Verification Method of Embedded Software Requirements Analysis Document," *Journal of Computers*, vol. 31, no. 6, pp. 210–229, 2020, doi: https://doi.org/10.3966/199115992020123106017

[5] K. Allenby and T. Kelly, "Deriving safety requirements using scenarios," in *Proceedings of the IEEE International Conference on Requirements Engineering*, 2001, pp. 228–235, doi: 10.1109/isre.2001.948563.

[6] G. Sindre, "A look at misuse cases for safety concerns," in *IFIP International Federation for Information Processing*, 2007, vol. 244, pp. 252–266, doi: 10.1007/978-0-387-73947-2_20.

[7] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, 2005, doi: 10.1007/s00766-004-0194-4.

[8] M. Damodaran, "Secure Software Development Using Use Cases and Misuse Cases," issued in *Information Systems*, vol. 7, no. 1, pp. 150–154, 2006, doi: 10.48009/1_iis_2006_150-154.

[9] National Highway Traffic Safety Administration, "Traffic Safety Facts Annual Report, 6/30/2020," Www-Nrd.Nhtsa.Dot.Gov, no. May, p. Volume: 2001, Issue: June, Pages: 232, 2014, Accessed: Dec. 27, 2020. [Online]. Available: https://cdan.nhtsa.gov/tsftables/National Statistics.pdf.

[10] Tesla, "Tesla Vehicle Safety Report," *Tesla Vehicle Safety Report*, 2020, Accessed: Dec. 27, 2020. [Online]. Available: https://www.tesla.com/VehicleSafetyReport.

[11] "ISO 25010.", Accessed: Dec.30, 2020. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=3&limitstart=0.

[12] A. Lyon, J. Coifman, H. Cook, F. Liu, K. Ludwig, and S. Dorsey, "The Cognitive Walk-Through for Implementation Strategies (CWIS): A pragmatic methodology for assessing strategy usability," in *11th Annual Conference on the Science of Dissemination and Implementation*, pp. 1–32, 2018, doi: https://doi.org/10.21203/rs.3.rs-136222/v1.

[13] P. A. Zielinski, "Concept of safety and safety requirements for dams," in *Dams and Reservoirs under Changing Challenges - Proceedings of the International Symposium on Dams and Reservoirs under Changing Challenges - 79 Annual Meeting of ICOLD*, Swiss: Swiss Committee on Dams, pp. 153–162, 2011, doi: 10.1201/b11669-22.

[14] D. G. Firesmith, "Engineering safety-related requirements for software-intensive systems," in *Proceedings. 27th International Conference on Software Engineering*, pp. 720–721, doi: 10.1109/ICSE.2005.1553680.

[15] D. Seifert, "Model-based Refactoring for Component Fault Trees," Accessed: Dec. 27, 2020. [Online]. Available: https://www.researchgate.net/publication/341121052_Model-based_Refactoring_for_Component_Fault_Trees.

[16] Silvianita, M. F. Khamidi, I. Rochani, and D. M. Chamelia, "Hazard and Operability Analysis (HAZOP) of Mobile Mooring System," in *Procedia Earth and Planetary Science*, vol. 14, pp. 208–212, Jan. 2015, doi: 10.1016/j.proeps.2015.07.103.

[17] G. Sindre and A. Opdahl, "Capturing security requirements through misuse cases," NIK 2001, Norsk Informatikkonferanse 2001, pp. 12, 2001, Accessed: Dec. 27, 2020. [Online].

[18] S. Caroline, "What is strenuous? Driving itself or the driving situation?," Jan. 2006, Accessed: Dec. 27, 2020. [Online]. Available: https://www.researchgate.net/publication/225018978_What_is_strenuous_Driving_itself_or_the_driving_situation.

[19] M. Werling, T. Gindele, D. Jagszent, and L. Gröll, "A robust algorithm for handling moving traffic in urban scenarios," in IEEE Intelligent Vehicles Symp., Proc., pp. 1108–1112, 2008, doi: 10.1109/IVS.2008.4621260.

[20] T. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik," in *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 1, p. 77, 2018, doi: 10.25126/jtiik.201851610.