SYSTEMATIC EVALUATION AND IMPROVEMENT OF SECURITY REQUIREMENT ENGINEERING NOTATIONS

BY

Faisal Saleh

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

May 2014

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Faisal Saleh** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

____
Dr. Mohamed El-Attar
(Advisor)

عادل
____
Dr. Adel Ahmed

Department Chairman

Sajjad Mehmod
____
Dr. Sajjad Mahmood
(Member)

____
Dr. Salam A. Zummo
Dean of Graduate Studies

____
Dr. Moataz Ahmed
(Member)

١٧/٦/١٤
____
Date

Dedication

I would like to dedicate this work to my parents, brothers and sister. This work would not be possible without their support. I would also like to thank all of my previous teachers, as their hard work has laid the foundation for this work.

ACKNOWLEDGMENTS

First I would like to thank almighty Allah for giving me the ability and the strength to work on and complete this thesis.

I would like to give my sincere appreciation to my advisor Dr. Mohamed El-Attar for to all the hard work he has put in this work, and for the guidance and encouragement he provided throughout this research. Dr. Attar has been a mentor, teacher and a friend. I feel very lucky to have had Dr. Attar as my thesis supervisor. I would also like to thank Dr. Attar for helping me in my future endeavors.

I would also like to thank the committee members, Dr. Moataz Ahmed and Dr. Sajjad Mahmood, for dedicating time out of their busy schedule for this work and providing their feedback.

Lastly, I would like to thank Mazin, for his help and providing some great entertainment with his stories. I will miss the tea/coffee breaks that were undertaken during our work on the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPMN	:	Business Process Modeling Notation
E	:	Errors Committed
MUCM	:	Misuse Case Maps
NN	:	New Notation
ON	:	Old Notation
PoN	:	Physics of Notations
T	:	Response Time
UC	:	Use Case
UCM	:	Use Case Maps
UML	:	Unified Modeling Language
VTML	:	Visual Traceability Modeling Language

ABSTRACT

Full Name	:	Faisal Saleh
Thesis Title	:	Systematic Evaluation and Improvement of Security Requirement Engineering Notations
Major Field	:	Computer Science
Date of Degree	:	May 2014

[Context and Motivation] Security has become one of the most important aspects of software design. Software in today's world needs to detect and stop threats posed from authorized and unauthorized users. Misuse case (MUC) and misuse case maps (MUCM) modeling notations allow security analysts to consider and account for security requirements in the early stages of a development process instead of relying on generic defensive mechanisms that are augmented to software systems post-development. **[Problem/Question]** Most research contribution in the area of MUC and MUCM focus on extending the notation to increase its coverage of additional security related semantics. However, there lacks research that evaluates the perception of MUC and MUCM models by its readers. A misread or misinterpreted model can have dire consequences downstream leading to the development of an insecure system. **[Principal Ideas]** This work presents a scientific evaluation of the cognitive effectiveness of MUC and MUCM modeling notation based on theory principles and empirical evidence from the cognitive science field. Such evaluations can be the basis on which the notations can be improved further. **[Contribution]** The evaluation of MUC and MUCM highlights several instances where the cognitive effectiveness notations in question can be improved. We have modified the MUC modeling notation based on the evaluation to improve its cognitive effectiveness. Furthermore, we have conducted an extensive empirical evaluation of the improvements that has given positive results.

ملخص الرسالة

الاسم الكامل : فيصل صالح

عنوان الرسالة: تقييم و تحسين منهج ي لرموز هندسه المتطلبات الآمنة

التخ صص: علوم الحاسب

تاريخ الدرجة العلمية: مايو- 2014

(المحتوى و الحافز) اصبح الأمن من أحد اهم الجوانب في تصميم البرمجيات. في الوقت المعاصر، تحتاج البرمجيات الى التعرف و إيقاف المخاطر الناجمه عن المستخدمين المصرح و غير المصرح بهم. رموز نمذجه حالات سوء الإستخدام و خرائط حالات سوء الإستخدام تسمح لمحللي الحمايه بنمذجه المتطلبات في الأنظمه خلال مراحل باكره من عمليه التطوير عوضاً عن الإعتماد على تقنيات حمايه عامه تضاف للأنظمه بعد الانتهاء من عمليه التطوير. **(المشكله/ السؤال)** العديد من البحوث أضافت الى مجالات نمذجه حالات سوء الإستخدام و الى خرائط سوء الإستخدام العديد من المعاني التي زادت قابليه تلك النماذج على احتواء اكبر قدر من المعاني و المفاهيم الأمنيه. عموماً، هنالك نقص في الابحاث التي تقيم الإستقبال البصري لقارئ نماذج حالات سوء الاستخدام و خرائط سوء الاستخدام. النماذج التي يساء فهمها او تفسيرها يمكن ان تؤدي نتائج كارثيه و التي تباعا تقود الى تطوير أنظمه غير آمنه. **(الفكره الأساسيه)** هذا البحث يقدم تقييم منهجي لفعاليه إدراك رموز نماذج حالات سوء الإستخدام و خرائط حالات سوء الإستخدام بالإعتماد على نظريات و دلائل مثبته من علم ا.لإبا .كارلإعتماد على مثل هذا التقييم يمكن تحسين مكامن القصور في تلك النماذج. **(المساهمه)** تقييم نماذج حالات سوء الإستخدام و خرائط حالات سوء الإستخدام أوضح عده أمثله على مكمن قصور تلك الرموز من ناحيه فعاليه الإدراك. قمنا بتعديل رموز نماذج حالات سوء الإستخدام بالإعتماد على تقييم فاعليه الإدراك ثم تباعا تحسين تلك الفاعليه. ايضا هذا البحث قام بعمل دراسه تظهر ان الاقتراحات قامت بتحسيذ نماذج سوء الإستخدام من ناحيه الاستجابه الزمنيه و من ناحيه الأخطاء المرتكبه من قبل المشاركين في البحث.

CHAPTER 1

INTRODUCTION

Progress in software development methodologies has paved the way to develop more complex systems. Security in such systems has become a key design quality attribute where a secure system should also have the ability to detect potential threats and act accordingly. Security threats can originate from outside users trying to harm a system and from insiders with authorized access. A secure system should have a mechanism to keep malicious users out and have access control measures for inside users.

Traditional requirement engineering modeling techniques lack support to specify security related issues. To this end, a number of modeling techniques have been devised to account for important security aspects. In this study, we will focus on two security based modeling techniques, *Misuse Case* (MUC) [1] and *Misuse Case Maps* (MUCM) [2]. Misuse case modeling notation is an extension of the popular Use Case modeling technique (UC) [3] and as a results, is used to model the functional security requirements of a system. Likewise, Misuse case maps is an extension of the popular Use Case Maps modeling technique (UCM) [4] that models a systems architecture and its behavior. Misuse case maps addressed security requirements with respect to the architecture of the system. It models the threats that are present on the system architecture.

Visual notations provide crucial means of conveying information between different stakeholders of different backgrounds. To accurately communicate information, notations need to be easily comprehensible by the human mind (cognitively effective) [5]. The afore mentioned notations are no exception, yet decisions regarding the design of graphical symbols for software engineering notations are made subjectively, without providing any insight on the selection process [6], [7], [8]. These design choices can affect the cognitive expressiveness of a notation and its ability to convey information in an informative manner. Research in the field of visual notations design in software engineering has discriminately focused on the semantic constructs that they are desired to support. It is important for a modeling language to support important semantics; however, it is arguably equally as important to support its readability and comprehension by its potential readers. Whilst a misread or misinterpreted functional requirement may lead to the development of a system that does not satisfy its functional requirements, a misread or misinterpreted security diagram may lead to the development of an insecure system, effectively rendering it useless.

Recent developments have allowed researchers to evaluate and design visually effective notations. In particular, the "Physics of Notations" (PoN) [6] defines principles that can be used to evaluate the cognitive effectiveness of the visual syntax of notations. These principles are based on theory and evidence from multiple fields. The focus of these principles is on the visual aspects of a notation rather than semantic aspects. The first goal of this study is to evaluate the MUC and MUCM modeling techniques and propose changes to MUC modeling that adhere to afore mentioned principles. The second goal is to evaluate whether the changes conforming to the principles for MUC modeling will result in an extension of the notation that is better equipped to support readability and comprehension by its readers.

1.1. Problem Statement

Focus on visual notations used during requirement engineering for readability and comprehension has been lacking until recently. PoN provides a way to systematically evaluate and improve visual notations. This study focuses on the following two questions:

- What are the shortcomings in MUC and MUCM modeling techniques according to PoN?
- Does resolving the shortcomings highlighted by PoN actually improve MUC?

1.2. Objectives

The main objective in this study is to evaluate and improve the effectiveness with which the MUC and MUCM modeling techniques convey information to users. The main objectives of this work are as follows:

1. Evaluate the MUC modeling notation using the principles defined in PoN and formulate suggestions that satisfy the principles wherever possible.
2. Evaluate the MUCM modeling notation using the principles defined in PoN.
3. Conduct an empirical evaluation on the proposed notation for MUC containing the changes suggested in step 1. The purpose of the evaluation is to determine whether the effects of the changes proposed are positive or negative when compared with the original notation.

1.3. Research Methodology

The research methodology in this works is as follows:

1. Literature Review – In this step, the principles defined in PoN were studied meticulously, followed by review of literature to find if the evaluation of MUC and MUCM using PoN has been undertaken already.
2. Evaluation of notations – Next, we conducting an evaluation of the MUC and MUCM modeling techniques using PoN, thus answering the first of the research question.
3. Formulate Suggestions – Based on the shortcomings highlighted in Step 2, we formulate suggestions for MUC modeling notation that adhere to and overcome the challenges found during evaluation.
4. Evaluation of proposed notation – Conduct a controlled experiment in which the aim is to evaluate the performance of proposed notation based on factors such as time and errors.
5. Result compilation – The results of the studies are compiled and studied to conclude whether the following the principles defined in PoN actually improves the MUC modeling notation, thus answer the second research question.

1.4. Contributions

The contributions of this study are evaluation of MUC and MUCM security modeling techniques according to the principles defined in PoN. This study also contains a proposed extension to the MUC modeling notation that has yielded better performance than the original notation during empirical evaluation.

1.5. Outline

The rest of the work is organized as follows. Chapter 2 presents a literature review of notations evaluated using PoN, followed by a brief introduction to PoN, MUC and MUCM. Chapter 3 contains the evaluation of MUC and MUCM using the PoN, highlighting problems in the notation that affect the cognition. In chapter 4, we first provide suggestions to improve the MUC notation and then present the results of empirical evaluation for the NN. Finally, Chapter 5 concludes the thesis and presents some directions in which future work can be done. We have also added all the information used to conduct the empirical evaluation and all the results of the evaluation at the end as appendix.

CHAPTER 2

LITERATURE REVIEW

The "Physics of Notations" [6] defines principles that are based on theory and empirical evidence from multiple fields. These principles focus on the visual aspects of a notation with the goal of analyzing, comparing and improving visual notations by improving their cognitive effectiveness. Several evaluations of existing notations have been undertaken with the principles as the theoretical basis.

An evaluation of UML diagrams family was presented in [9]. The study focused on the common elements present across the entire family of UML diagrams. The authors argue that visual development of UML diagrams is lagging because of lack of attention to visual aspects. According to [9], class diagrams have the worst visual representation. The authors of the work suggested general improvements, instead of specific diagram based improvements, that are applicable to all of the diagrams in UML. In [5], the authors evaluated the goal-oriented modeling notation i*. Several shortcomings in the existing notation were found during evaluation. The authors suggested several improvements that included the use of color, more perceptually directed symbols and redundant coding. An evaluation of BPMN (Business Process Modeling Notation) [10] was presented in [11]. BPMN aims to provide a notation understood by all stakeholders. The evaluation found several shortcomings according to the principles that hinder its comprehension by some of its stakeholders. The authors provided suggestions to improve the cognitive effectiveness of the notation. The visual notation of use case maps (UCM) [4] was analyzed in [12]. The evaluation found several common weaknesses and suggested improvements. One of the problems with the notation was the large number of graphical symbols used that create a significant load on the cognition of the user.

In addition to the evaluation of existing notations, the principles of cognitively effective visual notations have been used to design new notations that are readily comprehensible. The Visual Traceability Modeling Language (VTML) [13] was designed according to the principles in PoN. The aim of VTML is to define traceability strategies for a project and then visually represent trace queries as constraints upon subsets of the model.

The following section presents a brief introduction to the PoN evaluation framework that will be used as the basis for the evaluation of MUC and MUCM. The notation of MUC and MUCM are introduced afterwards.

2.1. Physics of Notation

The cognitive effectiveness of visual syntax in requirements engineering notations has been overlooked until recently. The seminal paper that brought the concept of evidence-based visual syntax evaluation of software engineering notation was only been published in 2009 [6]. The paper explained the "Physics of Notation" and the outcome is nine evidence-based principles upon which the cognitive effectiveness of a notation can be evaluated. The principles are based on theory and empirical evidence from various science fields, in particular the cognitive science field. The principles can, and should be, used as basis for improving current notation and when formulating new notations.

In order to introduce any framework with the ability to create and improve a notation, we must first understand how visual notations communicate, which requires help from other fields such as communication, semiotics, graphic design, visual perception and cognition. The underlying communication theory [14] involves the ***sender encoding*** a ***message*** that a ***receiver decodes***. In the context of visual notations, the diagram designer (sender) creates (encode) the diagram (message) that other stakeholders (receiver) interpret (decode). The underlying message is lost if any uncertainty is present while encoding and decoding.

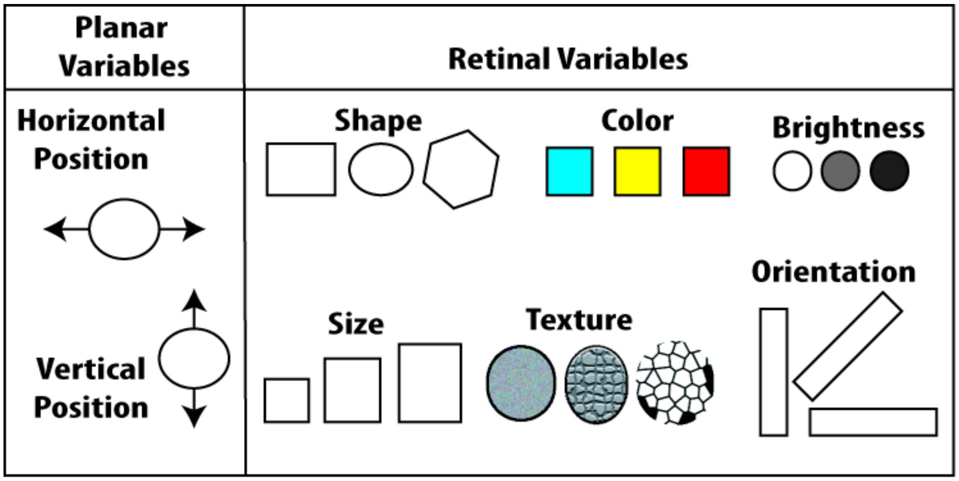


Figure 1 Bertin's Visual Variables for Graphical Notations

For words and sentences to be created in any language, certain primitive alphabets have to be defined. These alphabets are then combined in various forms to create words. To encode information graphically, Bertin [15] defined 8 visual variables (Figure 1): horizontal position, vertical position, shape, color, brightness, size, texture, and orientation. These variables are the visual alphabets for graphical notations, which can be combined in any combination to create unlimited amount of graphical symbols. Visual notation designers must use these variables to create the visual notation set that is most appropriate to their task. The visual notation set is an important aspect that must be considered for optimizing communication (encode and decode).

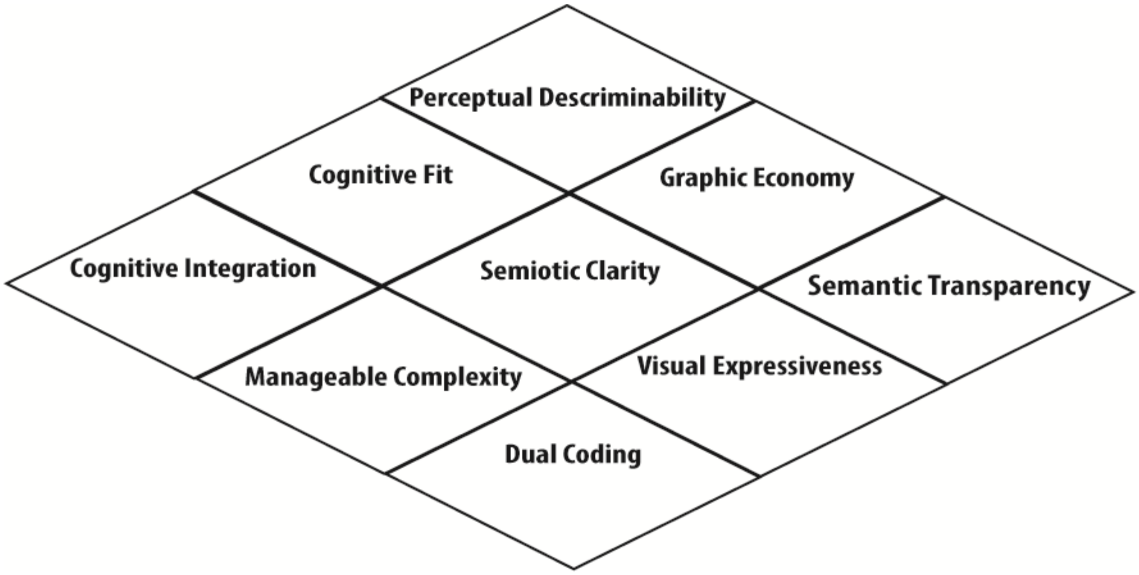


Figure 2 Principles of Cognitive Effective Visual Notation

Based on theories from other fields, the principles (Figure 2) in PoN focus on the perceptual attributes of a visual notation rather than the semantics of its graphical symbols. A brief summary of each principle is presented next.

2.1.1. Semiotic Clarity

The principles for designing of cognitively effective visual notations are presented as a modular structure at the center of which lies the principle of Semiotic Clarity. The principle of semiotic clarity states that notations should be designed in such a way that the graphical symbols have a one-to-one mapping to the semantic constructs they represent. Failure to achieve this desired mapping can lead to one or more of the following four anomalies (Figure 3).

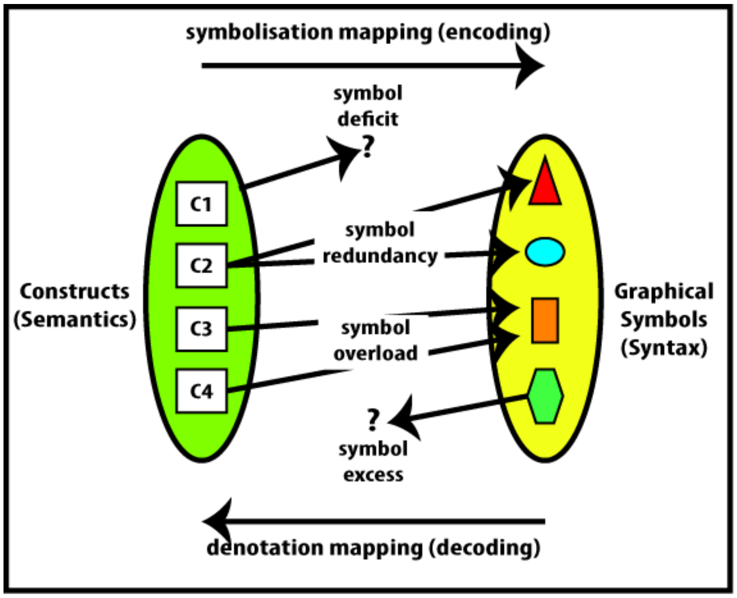


Figure 3 Principle of Semiotic Clarity

— *Symbol Redundancy* - Occurs when multiple graphical symbols are available to represent of one semantic construct. This can lead to confusion when using the notation, as although the author has a choice, there is no clear method to choose one over the other.

— *Symbol Overload* - Occurs when one graphical symbol represents multiple semantic constructs. Perhaps the most serious anomaly as a reader of the notation cannot definitively decide which construct a given symbol represents.

— *Symbol Deficit* - Occurs when a semantic construct is not represented by any graphical symbol.

— *Symbol Excess* - Occurs when a graphical symbol does not have a referent semantic construct.

Evaluating a notation for the principle of semiotic clarity requires finding the presence of the aforementioned anomalies by comparing the symbol set of the notation to its semantic constructs.

2.1.2. Visual Expressiveness

Recall that notation designers must use Bertin’s [15] visual variable to design visual notation set. The greater the number of visual variables used and the ranges used within each variable, the more visually expressive a notation becomes. The use of text is encouraged as a means for redundant coding that complements graphical symbols, instead of means to differentiate constructs. Using few visual variables can also have adverse effects on the perceptual discriminability of symbols within one notational set as discussed next.

2.1.3. Perceptual Discriminability

Perceptual Discriminability is concerned with the ease with which graphical symbols belonging to a notation can be differentiated from each other. It requires visual language designers to maximize their utilization of different visual variables to increase the visual distance [6] between symbols in order to visually stand out for easier recognition.

2.1.4. Dual Coding

Using Text as the primary means to distinguish between two constructs is ill advised [6], according to the principle of Dual Coding. However, using text per se should not be forbidden. In fact, according to dual coding theory [16], using graphics and text together to convey information is more effective than using either on their own. Therefore, text should be used as a form of redundant coding to complement graphics. According to communication theory, redundant coding reduces errors and counteracts noise [17]. The extra information in verbal form provides better clarification of the meaning of constructs.

2.1.5. Graphic Economy

The principle of Graphic Economy refers to having a suitable number of graphical symbols available for use in any notation. A large number of graphical symbols can lead to complex diagrams and a steep learning curve for novices. Research has introduced an upper limit of 6 categories for humans to discriminate between perceptually distinct alternatives [18].

2.1.6. Semantic Transparency

The principle of Semantic Transparency is concerned with the use of graphical symbols that imply the meaning of their corresponding constructs. For example, a lock symbol suggests protection of some kind while a symbol of a key suggests the ability to open some kind of protection. The use of such symbols speeds up the interpretation of diagrams and reduces the time needed for novices to learn the notation.

2.1.7. Complexity Management

The principle of Complexity Management recommends that a notation should include mechanisms to manage complex diagrams. The mechanisms enable models to convey information without overloading the human mind. Modularization and hierarchical abstractions are two techniques that can be used to manage complexity [6]. Modularization is concerned with the division of a large complex system into smaller parts that are easier to comprehend. Hierarchical abstraction focuses on representing a large complex system with different levels, where levels have different degrees of detail.

2.1.8. Cognitive Integration

The principle of Cognitive integration refers to the inclusion of mechanisms to support integration of information with other diagrams. Conceptual integration and perceptual integration are two categories of cognitive integration. Conceptual integration is concerned with providing mechanisms to help assemble information from multiple diagrams to form a more complete model mentally. Perceptual integration involves providing cues for easier navigation between multiple diagrams.

2.1.9. Cognitive Fit

The cognitive fit principle is concerned with having various suitable dialects for each targeted audience. Some dialects can be made complex and suitable for advanced users while other dialects can be simplified and made suitable for novices. Any requirements engineering artifact should be readable and comprehensible by non-technical stakeholders who can provide critical early phase feedback.

2.2. Misuse Cases

Sindre and Opdahl [19][20][21] first described misuse cases as negative use cases, highlighted by their unwanted interaction with a system. MUC modeling extended UC modeling by introducing four new concepts that are critical to specifying functional security requirements. The new concepts are *misuse cases, misusers, threaten* *relationship* and *mitigate relationship*. Their ensuing work [1] formally defined the concepts of a misuse case and misuser as follows:

- *Misuse Case* – “a sequence of actions, including variants, that a system or other entity can perform, interacting with misusers of the entity and causing harm to some stakeholder if the sequence is allowed to complete” [1]
- *Misuser* – “an actor that initiates misuse cases, either intentionally or inadvertently” [1]

The semantics of a misuse case and a misuser are similar, but inverse, to that of a use case and an actor, respectively. Hence, the graphical symbols for MUC were created by inverting the color scheme in UC notation while keeping the same symbols (see Figure 4). The *Threatens* relationship is used to indicate that a misuse case is threatening the integrity of a system while a particular use case is being executed. The *Mitigates* relationship is used to indicate that a use case is executed as a countermeasure to offset the attempted harm caused by a misuse case. Therefore, the Threatens relationship is depicted as an arrow that is directed from a misuse case to a corresponding use case. While the Mitigates relationship is depicted as an arrow that is directed from a use case to a corresponding misuse case.

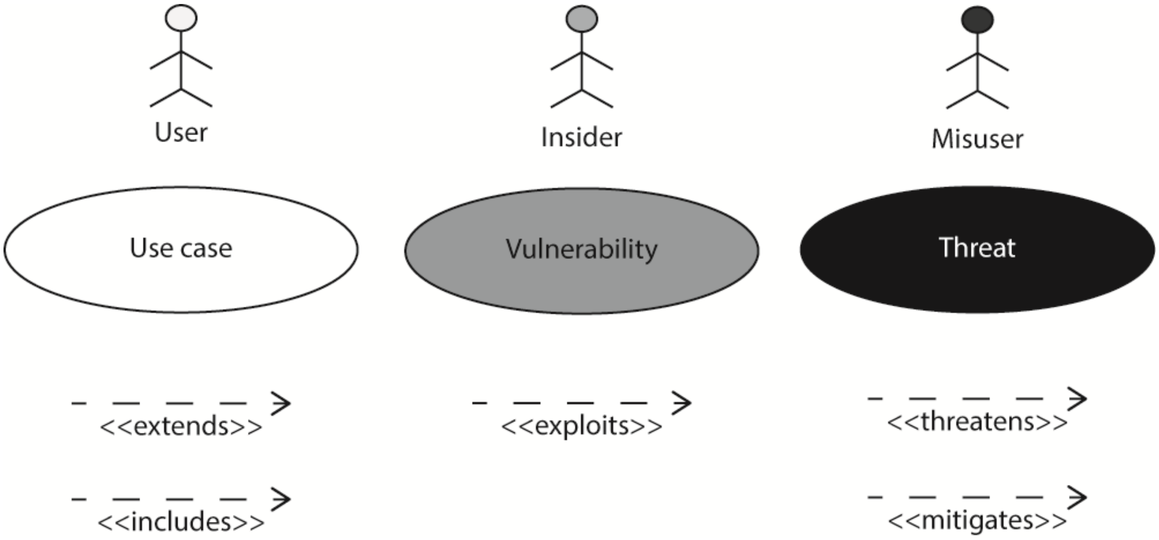


Figure 4 Misuse Case Notation Legend

Røstad [22] later expanded misuse cases to include the category of *insiders* that have different potential of attacking the system than misusers. An insider is a misuser who belongs to the group of authorized users for the system that is under attacked and hence has easier access to the system. Another addition by [22] to misuse cases was the addition of the *Vulnerability* and the *Exploits* relationship concepts. A *Vulnerability* is a weakness in the system that can be exploited by misusers (and insider) to attack the system. The *Exploits* relationship is used to denote which vulnerability a misuse case targets to harm the system with the use of a directed arrow from a misuse case to a vulnerable use case. Figure 4 contains the entire notational set of MUC that corresponds to the entire set of the aforementioned semantics.

2.3. Misuse Case Maps

Karpati, Sindre and Opdahl [2] saw a need for a requirement modeling technique that addressed security requirements with respect to a secure architecture. They extended the UCM [4] modeling technique with several constructs to help model security related scenarios to create the MUCM modeling notation. Key components of the extension include the addition of *exploit paths, vulnerability and mitigation*. Exploit paths are steps in a scenario that can compromise a system. Each exploit path can be numbered to show the steps in which a complex scenario can be achieved. Vulnerability is a weak point in the system where the exploit path intersects a component of the system. Misusers can exploit vulnerabilities to adversely affect the system, causing harm and disruption. Lastly, the system's ability to detect threats and counter can be modeled using mitigation.

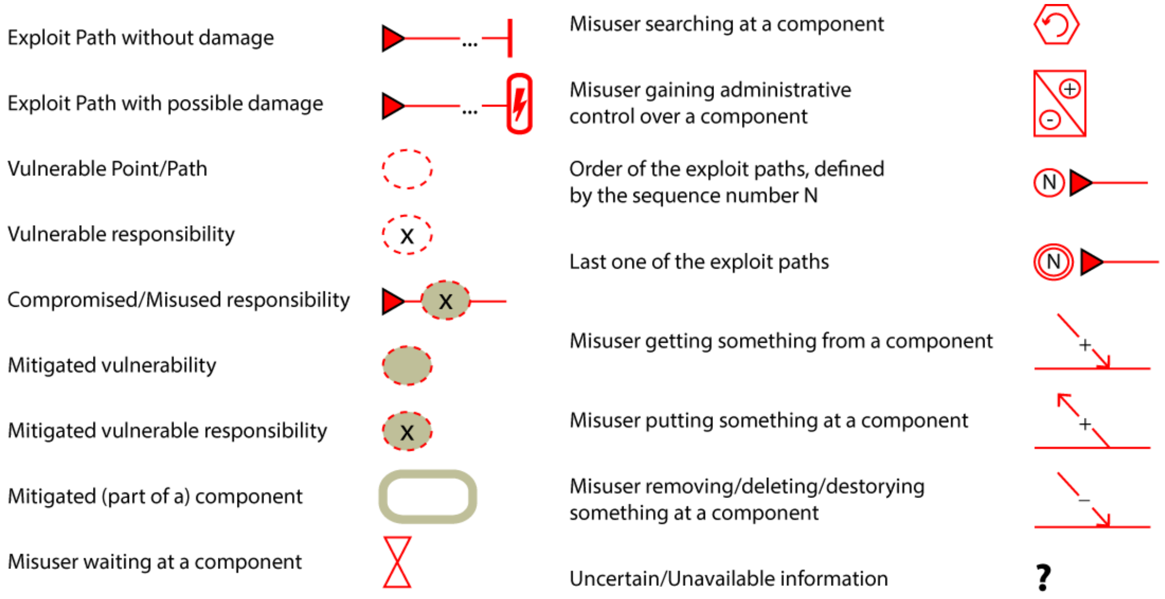

Exploit Path without damage
Exploit Path with possible damage
Vulnerable Point/Path
Vulnerable responsibility
X
Compromised/Misused responsibility
X
Mitigated vulnerability
Mitigated vulnerable responsibility
X
Mitigated (part of a) component
Misuser waiting at a component
Misuser searching at a component
Misuser gaining administrative control over a component
Order of the exploit paths, defined by the sequence number N
N
Last one of the exploit paths
N
Misuser getting something from a component
+
Misuser putting something at a component
+
Misuser removing/deleting/destorying something at a component
–
Uncertain/Unavailable information
?

Figure 5 Misuse Case Maps Modeling Legend

In addition to previously discussed components, several other components are also added. Figure 5 presents the full notation set of MUCM containing the security-based additions to UCM.

The authors of the paper provide a rationale on the choice of graphical symbol for the new components, albeit briefly. The authors explored using inverted symbols in the MUCM extension as was done to for the creation of MUC modeling extension. However, the original UCM notation’s use of filled symbols for start and end points along with solid lines for paths required the authors to use other means (color and shapes) to distinguish between positive and negative scenarios.

CHAPTER 3

NOTATION EVALUATION

3.1. Misuse Case Modeling

This section contains the evaluation results of the MUC notation according to nine principles defined in PoN [6]. The evaluation considers the original notational set introduced by Sindre and Opdahl [1] in addition to the expanded notational set introduced by Røstad [22]. The expanded notational set by [22] is considered in our evaluation as it is, according to the literature, the most advanced use case-based functional security-based modeling notation. The evaluation is presented in the following nine subsections, each summarizing a principle and presenting the evaluation results of the MUC modeling notation based on that principle. An outline of the subsequent subsections is presented in Figure 6.

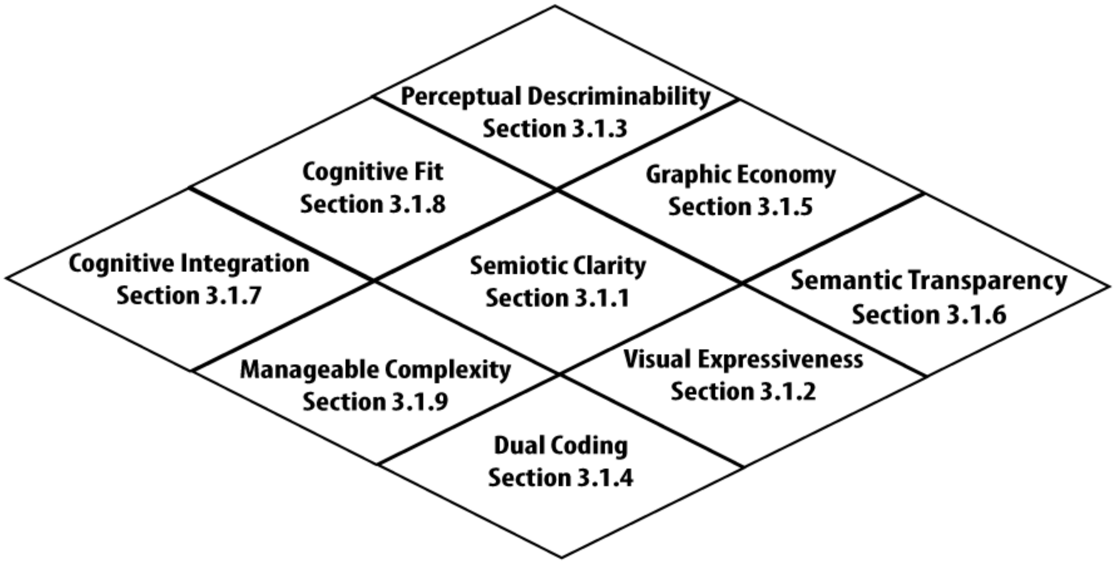


Figure 6 MUC Evaluation Outline

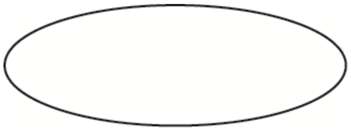
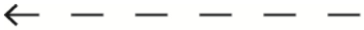
3.1.1. Semiotic Clarity

The principle of semiotic clarity states that notations should be designed in such a way that the graphical symbols have a one-to-one mapping to the semantic constructs they represent. Failure to achieve such mapping can lead to one of more of the following anomaly: Symbol Redundancy, Symbol Overload, Symbol Deficit, and Symbol Excess. Evaluating a notation for the principle of semiotic clarity requires finding the presence of the aforementioned anomalies by comparing the symbol set of the notation to its semantics. For MUC modeling notation, the semantics embodied by the technique derived from the representative literature, in particular the works of [1][22].

Symbol Redundancy - The MUC modeling notation does not contain any instances of symbol redundancy. All the constructs available in the meta-model have at most one corresponding graphical symbol to represent them.

Symbol Overload - The MUC modeling notation has two cases of symbol overload. Table 1 shows the symbols and their corresponding constructs that are involved in symbol overload cases. In UC modeling, the conventional oval symbol is used to represent a use case. However, in MUC modeling, these become specialized types of use cases: vulnerable, threatened and mitigating use cases. Vulnerable use cases have a unique grey background and thus are not involved in any symbol overload cases. However, threatened and mitigating use cases are important security based concepts that should be visually differentiated and not just semantically. One type is supposed to show potential weakness while the other should show abatement of a threatening misuse case. Threatened and mitigating use cases should also be differentiated visually from regular use cases. According to the current notation, regular, threatened or mitigating use cases use the same symbol.

Table 1 Overloaded symbols and their semantic constructs

Symbol	Semantic Constructs
	• Mitigating use case • Threatened use case • Regular use case
	• Includes relationship • Extends relationship • Exploits relationship • Mitigates relationship • Threatens relationship

The second case of symbol overload occurs with the arrow symbol which represents *includes*, *extends*, *exploits*, *threatens* and *mitigates* relationships. The dash arrows symbol is used to represent all of these relations. In the current notation, the only way to set apart relations is by reading the annotated textual stereotypes. The principle of dual coding (presented later in Section 3.1.4) advises against using text as the only means to distinguish between different symbols.

Symbol deficit and Symbol Excess - The MUC modeling notation has no symbol deficit since all the semantic constructs can be represented by one graphical symbol. Note that symbol deficit is evaluated based on the semantics described by the respective authors of the notations [1][22]. However, to fully prove the absence of any instances of symbol deficit, an in depth ontological analysis of the security domain will be required in order to identify any missing semantic concepts not covered in [20][22], which is beyond the scope of this work. Similarly, the MUC modeling notation does not contain any instances of symbol excess as each graphical symbol corresponds to at least one semantic construct.

3.1.2. Visual Expressiveness

The principle of Visual Expressiveness suggests that notation designers use as many of Bertin’s [15] visual variables to increase the expressiveness of notation as a whole, while limiting the use of text. The notation of MUC modeling only uses a combination of two visual variables: *shape* and *brightness*. For the shape visual variable, three different values are used: ovals, stickman figures and arrows. Brightness levels vary in each of the different shapes. Ovals and stickman figures have three brightness levels: white, grey, and black. Arrows only have two brightness values: solid and dashed. To compensate for the underutilization of visual variables, the MUC modeling notation makes extensive use of textual encoding. The use of text is discouraged except in the case of redundant coding that complements graphical symbols. Using such few visual variables can also have adverse effects on the perceptual discriminability of symbols within one notational set as discussed in the following section.

3.1.3. Perceptual Discriminability

Perceptual Discriminability is concerned with the ease with which graphical symbols belong to the same notation can be differentiated from each other. It requires visual language designers to maximize their utilization of different visual variables levels to increase the visual distance [6] between symbols in order to visually stand out for easier recognition. Recall that the MUC modeling notation only uses two visual variables: *shape* and *brightness*. Given that shape is the primary basis upon which humans identify objects in the real world [6][23][24][25] assessment of the MUC modeling notation with respect to the perceptual discriminability principle is performed by comparing symbols that belong to same shape category. Three shape categories are identified: ovals, stickman figures and arrows.

Ovals - This shape category includes regular, vulnerable, threatened and mitigating use cases, in addition to misuse cases. Recall that regular, threatened and mitigating use cases utilize the same exact symbol; therefore there is no perceptual discriminability between them. Brightness is the only means to distinguish between use cases, misuse cases and vulnerable use cases. Three levels of brightness are used: white for use cases, black for misuse cases, and grey for vulnerable use cases. The three levels of brightness used are appropriate for the semantics they represent. White and black are at the extreme ends of the brightness visual variable. This is appropriate as they are the complete opposites of one another with respect to security threat levels. Vulnerable use cases are not quite harmful as misuse cases yet they may eventually lead to damage being caused and therefore they are not quite peaceful as use cases. As such, the use of the color grey is also appropriate for vulnerable use cases since it is the midway value in the brightness range.

Stickman Figures - This shape category includes actors, misusers and inside attackers. Once again, brightness is the only means to distinguish between actors, misusers and inside attackers. Unlike the oval shape, brightness in stickman figures is only visible from the head part of a stickman figure. Therefore, perceptual discriminability between symbols within the stickman figure shape family is lesser than that of symbols within the oval shape family. If the size of the stickman figure is too small, especially the head, then brightness alone might be insufficient for the different concepts to be differentiated from one another, which may lead to misinterpretations. Once again the three levels of brightness are appropriately used with black and white denoting extreme ends of security threat levels. Inside attackers are not pure misusers as they are authorized users of a system, yet they intend harm and thus are not purely peaceful as actors. As such, the use of the color grey is appropriate for inside attackers.

Table 2 Relations with corresponding visual variable used to represent them.

Relationship	Symbol
Generalization	◁──────────
Association	──────────
Include, Extend, Exploit, Threatens and Mitigates	← ─ ─ ─ ─ ─
Directed Association	←──────────

Arrows - MUC modeling introduced three new relationships along with ones already present in UC modeling. The complete set of relationships is shown in Table 2. Symbols that appear in the same row cannot be differentiated visually. Once again, brightness is the only visual variable used to differentiate between symbols in different rows. Apart from the generalization relationship, the only way to differentiate between the other relations is through textual differentiation or by determining other syntax variables involved. For example, a directed association relationship can be differentiated from a threaten relationship without using text if it is connecting an actor and a use case. Similarly, a threaten relationship is only allowed to be directed from a misuse case to a use case. In general, relations in MUC modeling notation do not contain enough visual distance between them.

3.1.4. Dual Coding

The use of textual annotation as the primary means to distinguish between two constructs is ill advised [6] according to the principle of Dual Coding. However, using text per se should not be forbidden. In fact, according to dual coding theory [16], using graphics and text together to convey information is more effective than using either on their own. Therefore, text should be used as a form of redundant coding to complement graphics. According to communication theory, redundant coding reduces errors and counteracts noise [17]. The extra information in verbal form provides better clarification of the meaning of constructs. As discussed in previous sections, the MUC modeling notation heavily relies on textual differentiation as the only means to determine different relationships.

3.1.5. Graphic Economy

The principle of Graphic Economy refers to having a suitable number of graphical symbols available for use in any notation. A large number of graphical symbols can lead to complex diagrams and a steep learning curve for novices. Research has introduced an upper limit of 6 categories for humans to discriminate between perceptually distinct alternatives [18]. The MUC modeling notation satisfies the upper limit as the number of graphical symbols used in the notation fall into 3 distinct categories: ovals, stickman figures and arrows. It will be advised to leverage the extra room to introduce perceptually distinct symbols that can resolve aforementioned issues, such cases of symbol overload and low levels of perceptual discriminability.

3.1.6. Semantic Transparency

The principle of Semantic Transparency is concerned with the use of graphical symbols that imply the meaning of their corresponding constructs. For example, a stickman figure is widely understood to mean a person. The use of such symbols speeds up the interpretation of diagrams and reduces the learning curve for novices. In MUC modeling only the actor, misuser and attacker symbols, depicted as stickman figures, suggest the meaning of a person. However, it should be noted that actors, misusers and inside attackers might not be humans. In such cases the stickman figure may actually be more misleading than a neutral symbol such as the generic rectangle. The oval symbols used offer no suggestion of the interaction-based behavior they embody. Similarly, the various types of arrow symbols are not suggestive of the types of relationship they represent.

3.1.7. Cognitive Integration

The principle of Cognitive integration refers to the inclusion of mechanisms to support integration of information with other diagrams. Conceptual integration and perceptual integration are two categories of cognitive integration. Conceptual integration is concerned with providing mechanisms to help assemble information from multiple diagrams to form a more complete model mentally. Perceptual integration involves providing cues for easier navigation between multiple diagrams. Similar to UC models, MUC models do not contain any explicit mechanisms for conceptual or perceptual integration. This does not imply that techniques cannot be devised to facilitate cognitive integration. For example, a technique was introduced in [26] that can be used to transform MUC to mal-activity diagrams [27]. However, the technique enabled navigation from MUC to mal-activity diagrams via a model transformation algorithm and not via navigational features in the MUC modeling notation [26].

3.1.8. Cognitive Fit

The cognitive fit principle is concerned with having various suitable dialects for each targeted audience. Some dialects can be made complex and suitable for advanced users while other dialects can be simplified and made suitable for novices. A MUC model is a requirements engineering artifact. As is the case with any requirements engineering artifact, it should be readable and comprehensible by non-technical stakeholders who can provide critical early phase feedback. The MUC diagram notation does not contain multiple dialects. One of the most popular aspects of UC diagrams is its simplicity. The MUC diagram notation is not a major extension of the UC diagram notation. Therefore, it can be argued that the current MUC diagram notation is not expected to be very difficult to understand even by novices, as it is not a major extension to the UC diagrams notation.

3.1.9. Manageable Complexity

The principle of Complexity Management recommends that a notation should include mechanisms to manage complex diagrams. The mechanisms enable models to convey information without overloading the human mind. Modularization and hierarchical abstractions are two techniques that can be used to manage complexity [6]. Modularization is concerned with the division of a large complex system into smaller parts that are easier to comprehend. Hierarchical abstraction focuses on representing a large complex system with different levels, where levels have different degrees of detail. The MUC modeling notation contains no explicit mechanism for complexity management.

3.2. Misuse Case Maps

This section contains the evaluation results of the MUCM notation according to PoN. The evaluation considers the original notational set introduced by Karpati, Sindre and Opdahl [1]. The evaluation is presented in the following nine subsections, each summarizing a principle before presenting the evaluation results of the MUCM modeling notation based on that principle. An outline of the subsequent subsections is presented in Figure 7.

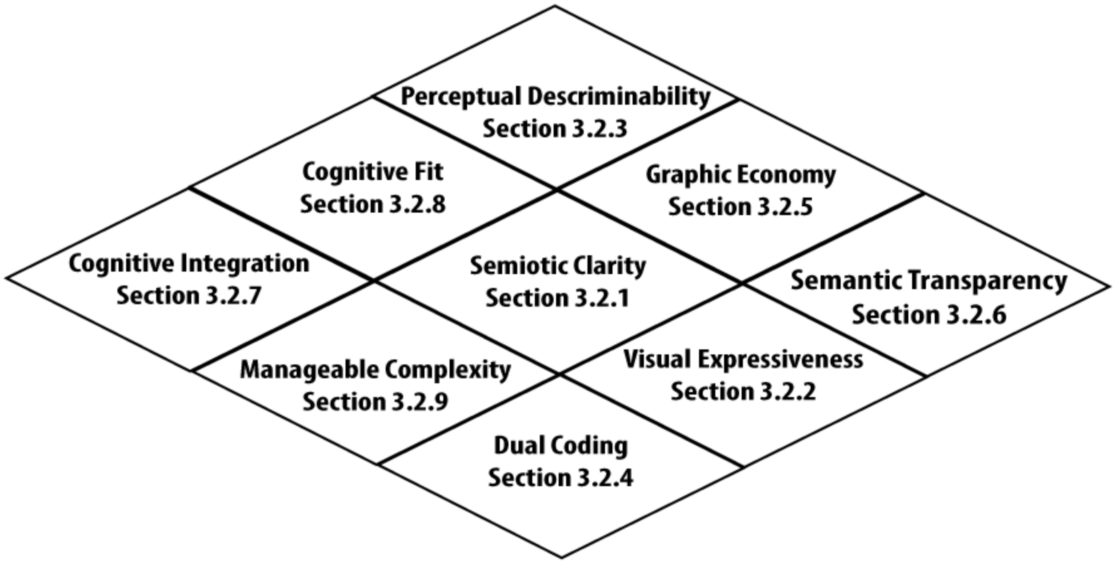


Figure 7 MUCM Evaluation Outline

3.2.1. Semiotic Clarity

The principle of semiotic clarity states that notations should be designed in such a way that the graphical symbols have a one-to-one mapping to the semantic constructs they represent. Failure to achieve such mapping can lead to one of more of the following anomaly: Symbol Redundancy, Symbol Overload, Symbol Deficit, and Symbol Excess. Evaluating a notation for the principle of semiotic clarity requires finding the presence of the aforementioned anomalies by comparing the symbol set of the notation to its semantics.

Symbol Redundancy - The MUCM modeling notation does not contain any instances of symbol redundancy. All the constructs available in the meta-model have at most one corresponding graphical symbol to represent them.

Symbol Overload - The MUCM modeling notation does not contain any instances of symbol overload. All the symbols used in the notation are used to represent at most one construct.

Symbol deficit and Symbol Excess - The MUCM modeling notation has no symbol deficit since all the semantic constructs can be represented by one graphical symbol. Note that symbol deficit is evaluated based on the semantics described by the respective authors of the notation [1]. Similarly, the MUCM modeling notation does not contain any instances of symbol excess as each graphical symbol corresponds to at least one semantic construct.

3.2.2. Visual Expressiveness

The principle of Visual Expressiveness suggests using as many of Bertin's [15] visual variables to increase the expressiveness of notation as a whole, while limiting the use of text. The notation of MUCM modeling has visual expressiveness of level 4 as it uses a combination of four visual variables: *location*, *color*, *shape* and *brightness*.

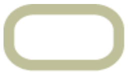
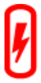
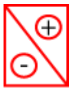
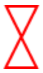
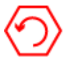
Location - MUCM uses the complete set of planar variables from Bertin's visual variables; the position of component on the diagram is crucial to conveying accurate information.

Color - The MUCM notation uses two values of color for its graphical symbols. The color red is used when documenting exploits paths regardless of their affinity to cause damage, while the color black is used to signify normal scenario paths.

Shape - The MUCM extension has introduced multiple new graphical symbols that are of different shapes (Table 3). The use of multiple new shapes has advantage of reducing the anomalies that can occur as specified in principle of semiotic clarity; however the use of excessive graphical symbols requires the reader to be better acquainted with the notation set, thus making the learning curve steeper.

Brightness - The security-based extension of MUCM varies brightness to signify mitigation. The vulnerable component is unfilled in the case of no mitigation, while it is fill gray if the component vulnerability has been mitigated.

Table 3 Shape categories of MUCM

Shape Category	**Graphical Symbol**
Triangle	
Oval	
Oval with lighting bolt	
Rectangle	
Hourglass	
Hexagon	

3.2.3. Perceptual Discriminability

Perceptual Discriminability is concerned with the ease with which graphical symbols belong to the same notation can be differentiated from each other. It requires visual language designers to maximize their utilization of different visual variables to increase the visual distance [6] between symbols in order to visually stand out for easier recognition. Recall that the MUCM modeling notation introduces multiple new shapes to represent different constructs. This makes the MUCM modeling notation to have a high level of perceptual discriminability, with a disadvantage of making the notation more difficult (as will be discussed in section 3.2.5).

3.2.4. Dual Coding

The use of text as the primary means to distinguish between two constructs is ill advised [6] according to the principle of Dual Coding. Therefore, text should be used as a form of redundant coding to complement graphics. The extra information in verbal form provides better clarification of the meaning of constructs. The MUCM modeling notation correctly uses text in its diagram to provide details regarding scenarios, path, and components. The use of text in this capacity is encouraged as it provides diagram readers with more information.

3.2.5. Graphic Economy

The principle of Graphic Economy refers to having a suitable number of graphical symbols available for use in any notation. A large number of graphical symbols can lead to complex diagrams and a steep learning curve for novices. Research has introduced an upper limit of 6 categories for humans to discriminate between perceptually distinct alternatives [18]. The MUCM modeling notation easily breaks this upper limit even when the only the security based additions are counted (Table 3). Since the MUCM notation incorporates the UCM modeling notation as well, which has graphic complexity of 28 [12], the number of graphical symbols needed to be accustomed to goes way beyond the upper limit defined by [18]. Such large numbers of graphical symbols will cause cognitive overload during interpretation and may lead to misinterpreted diagrams.

3.2.6. Semantic Transparency

The principle of Semantic Transparency is concerned with the use of graphical symbols that imply the meaning of their corresponding constructs. Of the six symbols introduced in MUCM (Table 3), only two can be considered semantic transparent. The hourglass symbols used to denote a misuser waiting at a component can be considered to signify a countdown timer. The second symbol is the lighting symbol denoting damage caused by an exploit scenario. The symbol of the lightning bolt is widely considered to represent danger. All other graphical symbols used conventional shapes that have no semantic meaning on their own.

3.2.7. Cognitive Integration

The principle of Cognitive integration refers to the inclusion of mechanisms to support integration of information with other diagrams using either conceptual integration or perceptual integration. MUCM modeling notation does not contain any explicit mechanisms for conceptual or perceptual integration.

3.2.8. Cognitive Fit

The cognitive fit principle is concerned with having various suitable dialects for each targeted audience. Some dialects can be made complex and suitable for advanced users while other dialects can be simplified and made suitable for novices. The MUCM notation does not contain any mechanism to simplify the resulting document. However, the intended target audiences of MUCM modeling diagrams are experts who are well professed in system architect design, thus removing the need of having different dialects for different level of users.

3.2.9. Manageable Complexity

The principle of Complexity Management recommends that a notation should include mechanisms to manage complex diagrams using either modularization or hierarchical abstractions. MUCM notation, like is counterpart UCM modeling notation, does not contain any mechanism to mange complexity. However, we believe the notation does contain a possibility to use hierarchical abstraction as a means to manage complexity. The top level will contain the full view of the system architect and the scenario paths, while the lower level can provide more detail in each component, thus reducing complexity and making it easier for user to interpret the diagram.

CHAPTER 4

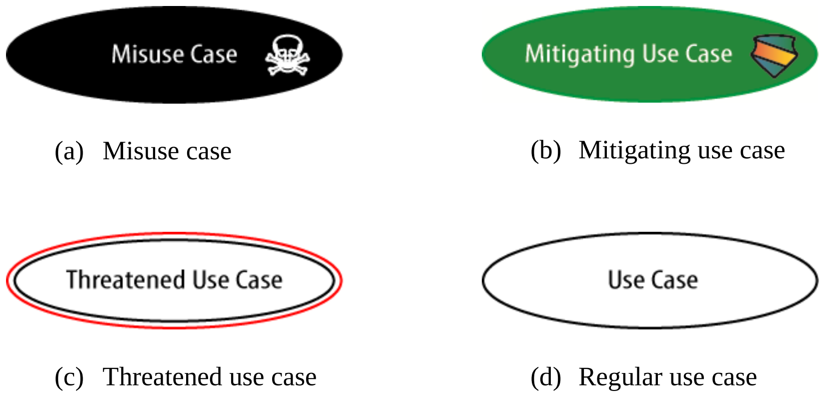
IMPROVING MISUSE CASES

4.1. Suggestions for MUC

The evaluation results presented in chapter 3 highlights a number of issues that makes the cognitive effectiveness of the MUC diagram notation suboptimal. In this section we provide suggestions to improve MUC. The improvements are suggested in line with the principles of designing cognitively effectiveness notations, which were used to evaluate the current notation. Note that our suggestions do not cover symbols from the UC modeling notation as it shares symbols with other UML diagrams. Therefore, suggesting changes to the use case modeling notation would require an analysis of the entire UML notational set, which is beyond the scope of this work. Our suggestions are as follows:

4.1.1. Case Nodes

One of the problems encountered with the original oval symbol used to represent the use case construct resulted in symbol overload as there are two other subtypes of use cases: mitigating and threatened use cases. It would ostensibly seem correct to introduce new shapes to represent the different types of use cases. However, given the strong influence of shape in cognition, care has to be taken before introducing new symbols as different shape categories are comprehended by humans to denote categorically different semantics [6]. As such, using new shapes to depict different types of use cases will imply that they are semantically very different from regular use cases, which is not the case. To differentiate between the types of use cases, we suggest using color. For threatened use cases, we suggest adding a second border that is colored red (Figure 8c), as the color red denotes warning/danger according to findings from the Color Psychology field [28]. The advantage of using adding a second border becomes apparent when a use case is threatened and mitigates a misuse case. Similarly the background color for mitigates use case can be changed to green, and shield icon can be added (Figure 8b). The shield icon is commonly used in software applications due to its semantic resemblance to defensiveness and safety. A bones and skeleton icon can be added the current symbol for misuse cases to annotate the new misuse case symbol (Figure 8a). The bones and skeleton figure, commonly present in the "Jolly Roger" flag, is a commonly known sign of imminent attack and intent to harm. All the changes introduced to the case nodes in MUC notation do not contain any changes to the actual shape of case nodes, which remains the same as in UC modeling (Figure 8d).



(a) Misuse case

(b) Mitigating use case

(c) Threatened use case

(d) Regular use case

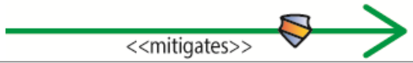
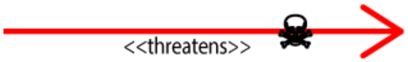
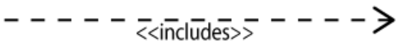
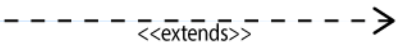
Figure 8 Suggested Graphical Symbols for Case Nodes

4.1.2. Relationships

Relationships in the MUC modeling notation do not have significant visual distance between them, as they do not use many visual variables. This leads to low levels of perceptual discriminability amongst the various types of relationships. To make matters worse, many relationships use identical graphical symbols leading to many cases of symbol overload. We suggest complementing relationships arrows with icons and using colors while keeping the textual stereotypes as a form of redundant encoding. We also suggest using different values of the *size* variable, meaning different arrows will have different thicknesses. Suggested improvements to the relationships notations are as follows (see Table 4):

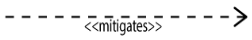
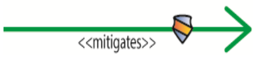
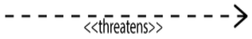
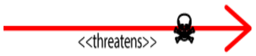
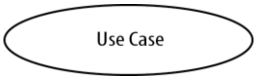
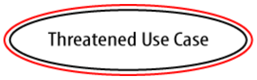
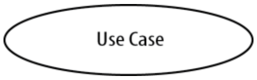
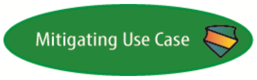
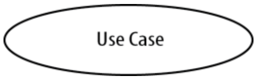
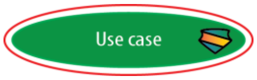
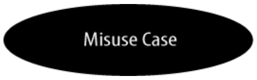
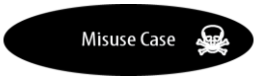
- *Threatens relation* – A thicker red colored arrow, annotated with a skeleton and bones icon. Use of the skeleton and bones icon is to be in line the suggested improvement for the misuse case notation. A thicker line suggests importance while the color red is suggestive of the danger is represents.
- *Mitigates relation* – Thicker green arrow, annotated with a shield icon. Use of the shield icon is in order to be in line with the icon used to annotate mitigating use cases. A thicker line suggests importance similar to the threaten relationship. Use of the color green is in order to be in line with the suggested improvement for the mitigating use case notation.

Table 4 Suggested Graphical Symbols for relationships

Relationship	**Suggested Notation**
Mitigate	
Threaten	
Include	
Extend	

In fact, mitigate and threaten relationships have a similar thickness level that is different from other relationships. This is due to the fact that they belong to security domain while highlighting the importance of specifying the security-related semantics. The complete set of improvements for MUC is present in Table 5. For each semantic construct, a brief justification for suggested notation is given along with the list of PoN principles that have been affected.

Table 5 Principles affected during improving MUC

Semantic	Old Notation	Suggested Notation	Justification	PoN Effected Principles
Mitigates relation			- Use shield symbol and green color to increase perceptual discriminability, keeping text as form of redundant coding	- Visual expressiveness - Perceptual Discriminibitliy - Dual Coding - Semiotic Clarity
Threatens relation			- Use skeleton symbol and red color to increase perceptual discriminability, keeping text as form of redundant coding	- Visual expressiveness - Perceptual Discriminibitliy - Dual Coding - Semiotic Clarity
Threatened Use case			- Use extra red border to reduce symbol overload	- Visual expressiveness - Perceptual Discriminibitliy - Semiotic Clarity
Mitigating Use case			- Use shield symbol and green background color to reduce symbol overload	- Visual expressiveness - Perceptual Discriminibitliy - Semiotic Clarity
Threatened and Mitigating Use case			- Use extra red border, shield symbol and green background color to reduce symbol overload	- Visual expressiveness - Perceptual Discriminibitliy - Semiotic Clarity
Misuse Case			- Use skeleton symbol to increase visual distance	- Perceptual discriminability

4.2. Empirical Evaluation

This section presents a subject-based experiment that was undertaken to examine the performance of the proposed NN with respect to the ON in light of two dependent variables. The experiment was performed at King Fahd University of Petroleum and Minerals, while the subjects used were enrolled in Software Engineering undergraduate degree. The experiment is reported using the standard experimentation process presented by Wohlin et al. [29].

To measure the performance and the effects of changes to MUC notation, two dependent variables were recorded. The dependent variables are shown in Table 6, along with their respective hypotheses. The first variable is the response time variable (*T*), for which the alternative hypothesis indicates that the time taken to interpret the diagrams developed using the NN will be less than the time required to interpret diagrams developed using ON. The second variable is the errors committed variable (*E*), for which the alternative hypothesis states that interpreting the diagrams developed using the NN will result in subjects committing fewer errors than interpreting diagrams developed in ON.

Table 6 The dependent variables and their corresponding hypotheses

Dependent Variable	**Null Hypothesis (Ho):**	**Alternative Hypothesis (Ha):**
Response Times	(Ho1): T (NN) ≥ T (ON)	(Ha1): T (NN) < T (ON)
Errors Committed	(Ho2): E (NN) ≥ E (ON)	(Ha2): E (NN) < E (ON)

The test subjects used in the experiment Software Engineering undergraduate students enrolled in the second year of their degree. The experiment was conducted during the second semester of 2014 academic year, i.e. Winter Session. The participating subjects had already taken an introductory Software Engineering course that covered UC modeling. Previous knowledge of UC modeling was an advantage for the study as the subjects were familiar with a notation set that is similar to the notation under analysis, thus reducing the overall learning curve. The subjects were given three lectures that covered the subjects of UC (refresh memory of students) and MUC in detail. In a subsequent session, the subjects performed exercises that further strengthen their knowledge of the MUC modeling.

During the experiment, subjects were asked to interpret MUC diagrams of two systems, an RFID-based product authentication system [30] (Figure 14 and 15) and a Swiss portfolio management company system [31] (Figures 12 and 13). The diagrams are presented in Appendix A. To prevent bias, it is beneficial not to use diagrams that were created by the authors. The MUC diagrams from [31] and [30] were selected for this experiment as they represent real systems. Slight editions to the diagrams were performed for two purposes: (1) to ensure that the entire notational sets were used in each diagram, (2) to approximate their structural characteristics. The diagrams were also designed to have the same length and width.

To perform the experiment, the subjects were divided evenly into two groups of 17 each (Group A and Group B). Two different diagrams were provided to the subjects where one of the diagrams was developed using the ON while the other diagram was developed using NN. For group A, the second diagram was developed using the NN, while group B had the first diagram developed using NN. Research suggests that layout of a diagram affects the graph comprehension [32]. Therefore, both versions of each diagram (NN and

ON) were developed using an identical layout. A questionnaire (presented in Appendix B) was provided to the subjects that asked questions pertaining to the diagrams, such as “Identify all **mitigating** use cases”. The questionnaire contains two identical sets of questions, one for each diagram. Legends for both NN and the ON were also provided to the subjects. The subjects were asked measure the time taken to answer question pertaining to one diagram.

To analyze the results of the experiment, two statistical tests were used: Mann-Whitney U statistic [33] and cliff’s delta [34][35][36]. Both of the tests are non-parametric, given the non-normal nature of the datasets. Mann-Whitney U is a non-parametric test that allows two hypotheses to be compared. The Mann-Whitney U was used to test the differences between the medians of related samples. The Hodges-Lehman method was used to calculate the confidence intervals around the difference between medians given at the standard $p<0.05$ level [37]. The second statistical test used is the Cliff’s delta, which is a non-parametric effect size measure. Cliff’s delta has been preferred over other statistical tests since empirical evidence suggests that it is superior when the data is non-parametric and possesses variance heterogeneity [38][39].

4.3. Analysis and Interpretation

In this section we investigate the effect of using the proposed MUC notation in comparison to the ON. This analysis explores the experimental results to provide some illustrative experiment-wide numerical analysis by considering the aggregated results from both MUC diagrams. The purpose of this analysis is to gain a more general assessment of the overall impact of using the NN and to provide additional confidence to accept or reject the hypotheses relating to (*T*) and (*E*) variables.

Figure 9 presents the cumulative performance of groups A and B with respect to the response time variable, while Figure 10 presents the cumulative performance with respect to the errors committed variable. Descriptive statistics about the two groups' performances are shown in Table 7. The results of the Mann-Whitney test (Table 8) indicate a statistical significance for both variables. The result of Cliff's delta calculations is presented in Table 9. The statistical significance for the response time variable is confirmed by a confidence interval range, which includes positive values only, as expected. However, the confidence interval range for the errors committed variable includes the number zero. This is in contract to the significance shown in the Mann-Whitney U test. These calculations lead us to accept the hypothesis related to the (*T*) variable while rejecting the hypothesis related to the (*E*) variable, pending other revelations from subsequent analysis.

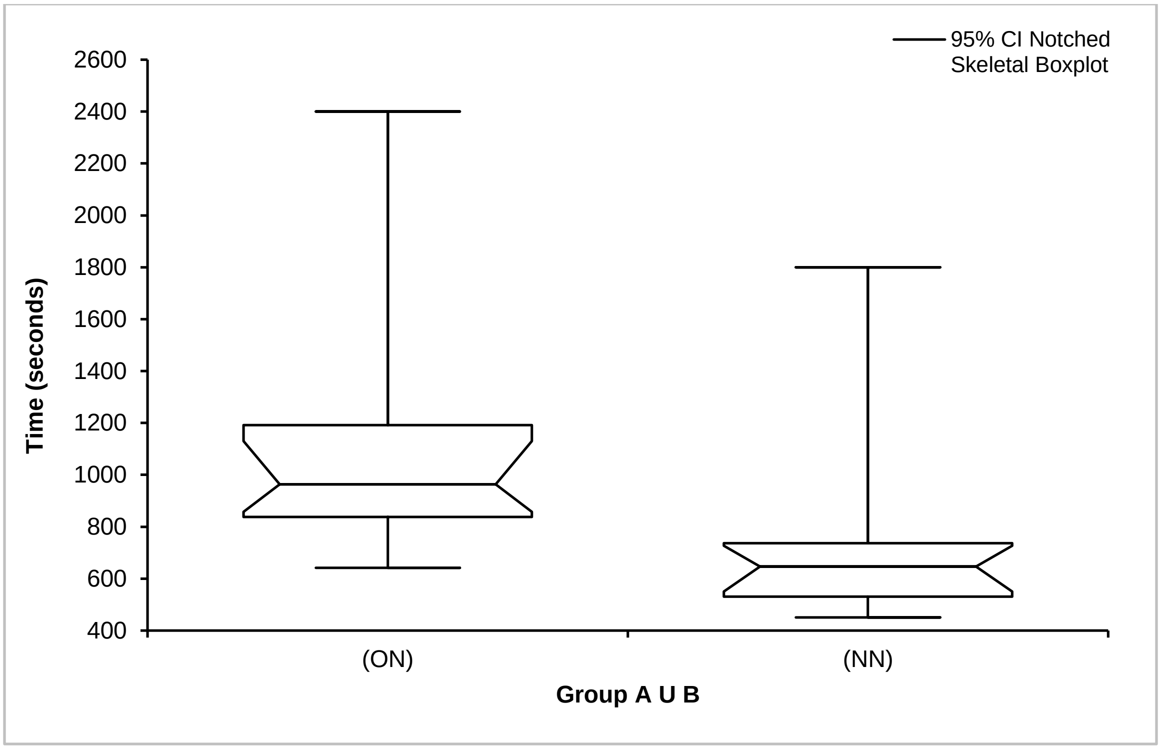


Figure 9 The cumulative performance of groups A and B with respect to response times for both misuse case diagrams

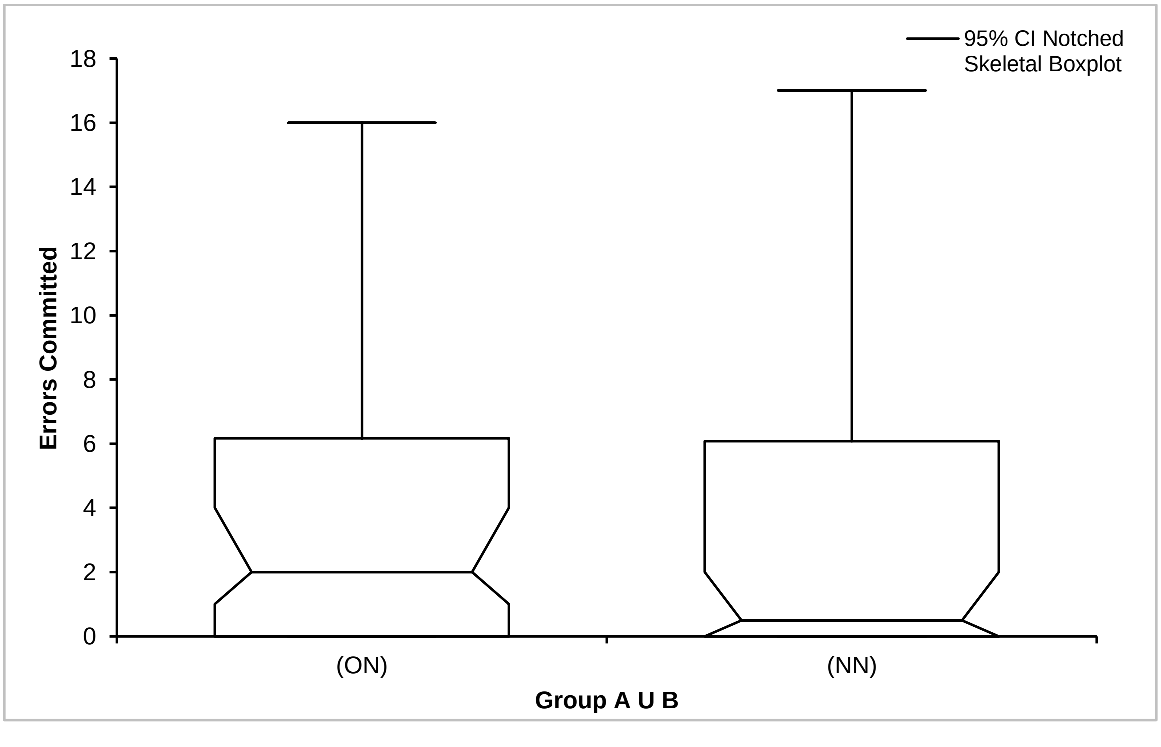


Figure 10 The cumulative performance of groups A and B with respect to errors committed times for both misuse case diagrams

Table 7 Descriptive statistics for Group A + B (samples = 34)

Variable	Notation	Min	1st Quartile	Median	3rd Quartile	Max	IQR
Response Times	NN	450.0	530.8	647.0	736.6	1800.0	205.8
	ON	642.0	837.8	963.0	1192.1	2400.0	354.3
Errors Committed	NN	0.0	0.0	0.5	6.1	17.0	6.1
	ON	0.0	0.0	2.0	6.2	16.0	6.2

Table 8 Mann-Whitney test for Group A + B (samples = 34)

Variable	Technique	Rank sum	Mean rank	U	Difference between medians	95% CI	Mann-Whitney U statistic	p
Response Times	NN	712.00	20.94	1039.00	328.00	257.00 to +∞	117.00	**<0.0001**
	ON	1634.00	48.06	117.00				
Errors Committed	NN	1034.00	30.41	717.00	1.00	0.00 to +∞	439.00	**0.0393**
	ON	1312.00	38.59	439.00				

Table 9 Cliff's delta for Group A + B

Variable	Cliff's delta ($\hat{\delta}$)	Variance	Confidence Interval around delta ($\hat{\delta}$)	
			maximum	**minimum**
Response Times	0.798	0.008	**0.914**	**0.562**
Errors Committed	0.240	0.019	**0.482**	**-0.035**

4.4. Qualitative analysis

In addition to the questions pertaining to the diagrams, the subjects were also provided a set of questions to elicit qualitative information after finishing questions pertaining to the diagrams. The three qualitative questions asked are as follows:

1. Which notation did you prefer?
2. What aspects of the notation you selected from (1) did you like?
3. What aspects of the notation you selected from (1) did you not like or feel that they can be improved?

Time used to answer qualitative questions was not recorded. These questions were added to help us get a better understanding of the preference of diagram users and the reasons for their choice.

For the first question, all subjects who responded to the question preferred the NN with the exception of two subjects, one in each group. The subject in group A was indifferent to either of the notations, while the subject in group B preferred the ON. For the second question, all subjects who preferred the NN in question 1 were happy with the use of color, as it made reading the diagrams easy. The significant cause of this is that the use of color makes graphical symbols more prominent, therefore easier to spot. The subject in group A, who was indifferent did not answer the question, while the subject who preferred the ON in group B responded that the use of color was the reason for their choice on Question 1. For the last question, which asked for suggestion/comments, the majority of the subjects suggested to modify includes and extends relation to make them more prominent. However in this study, the extends and the includes relations were not changed as these relations are part of widely used UC modeling notation, any changes to which would require in depth analysis of UML.

4.5. Limitations

The improvements proposed to MUC modeling in section 4.1 were effective at reducing the interpretation time, as evident during the empirical evaluation. The proposed changes to MUC depend on an increased use of color and graphical symbols. It can be said that the use of color will limit the usage scenarios of MUC modeling. The original MUC modeling notation is simple and the use of which requires little to no effort. The lack of color usage allows the diagrams to be constructed easily and quickly. However, in the proposed notation, the use of color is abundant and essential to the diagrams. This can limit the diagram creators, as additional tools are required some of which can lead to increased costs (color printing). However, it should be noted that the proposed notation introduces two changes to each semantic construct, color and graphical symbol. If the diagram developed using the NN is printed in grayscale (Figure 11), the graphical symbols introduces can be used to identify the semantic construct. The following identification technique can be used to identify the constructs:

- *Threatened case* – Two borders.
- *Mitigating case* – Case with shield symbol.
- *Misuse case* – Case with bone and skull symbol.
- *Regular case* – Case with no filling and one border.
- *Mitigates relation* – Solid arrow with shield symbol.
- *Threatens relation* – Solid arrow with bone and skull symbol.

With the aforementioned identification rules, it can be seen that the addition of symbols along with the use of color allows the diagrams developed using the NN to be interpreted even when color is not available. The limitation of using colors has been offset with the addition of symbols and given readers multiple ways to interpret the diagrams.

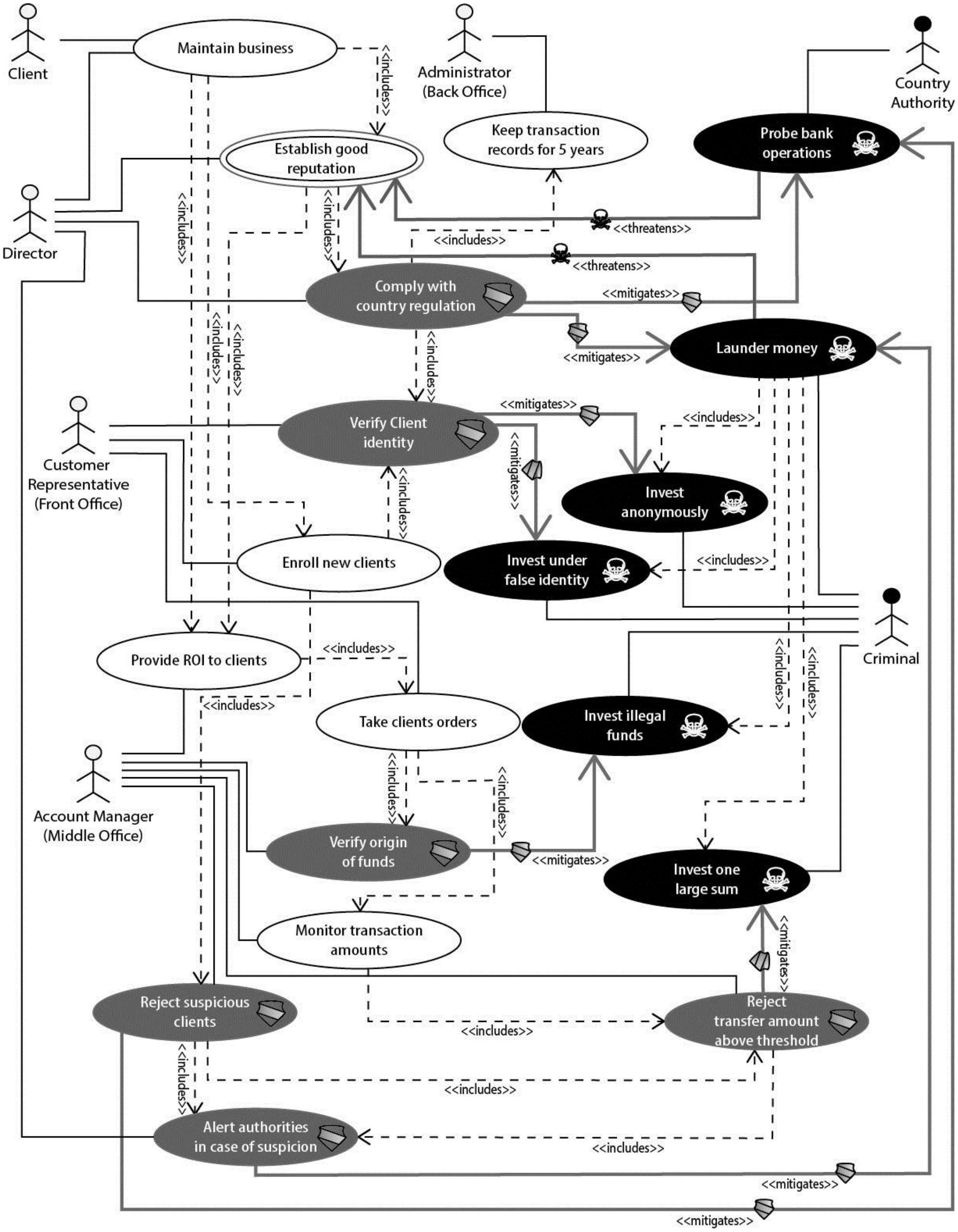


Figure 11 Proposed Notation Diagram in Grayscale

CHAPTER 5

CONCLUSION

In this work, we have undertaken a systematic evaluation of the MUC and MUCM modeling notation. The evaluation is performed on the visual syntax of the notation using the principles defined in PoN. In MUC, several issues with the current notation have been determined, which include symbol overload, extensive use of textual encoding as a means to differentiate graphical symbols, low perceptual discriminability between symbols and low visual richness in the symbols used. MUCM also contains several issues that include a large number of graphical symbols used (making the notation complex), low semantic transparency and no mechanism for complexity management. These issues can cause the diagrams to be misinterpreted, leading to insecure systems that are effectively rendered useless.

Following the evaluation, we have suggested improvements to MUC modeling notation to overcome the drawbacks highlighted. The modifications include the use of color, size and other graphical symbols. We have also changed the role of textual encoding in the new graphical symbols to complement the accompanying symbol and serve the purpose of redundant coding. The use of color to improve MUC is debatable for two reasons. First, color has a certain downside if the models in question are to be drawn by hand or printed. The lack of color utensils for drawing and printing will be in issue. Second, since MUC modeling is to be done alongside UC modeling, any attempt to significantly change MUC will affect the cohesion between the two modeling notations.

The improvements suggested in this paper were validated by user-studies. The user study is concerned with validating that the proposed notation can be read quicker and more accurately than the original notation. The user study was performed as an experiment that used software engineering students as subjects. There were two hypotheses set for the two variables of the experiment; response time and errors committed. The final hypotheses evaluation is shown in Table 10.

Table 10 The dependent variables and their corresponding hypotheses

Dependent Variable	**Null Hypothesis (Ho):**	**Alternative Hypothesis (Ha):**	**Results**
Response Times	(Ho1): T (NN) ≥ T (ON)	(Ha1): T (NN) < T (ON)	**Accepted**
Errors Committed	(Ho2): E (NN) ≥ E (ON)	(Ha2): E (NN) < E (ON)	**Rejected**

The main finding of the experiment is that the new notation can be read significantly quicker than the original notation. However, the results of the experiment did not allow us to accept the hypothesis that the new notation leads its reader to commit fewer reading errors. We believe that this phenomenon is due to the subjects spending more time reading the diagrams in order read it correctly. However, this conjuncture will require further empirical evidence.

Qualitative data obtained from the subjects indicates that all subjects finished the experimental tasks without facing fatigue or maturation issues and without facing time pressure. Therefore, we believe that the results obtained in this experiment are solely influenced by the treatments. Qualitative data obtained has also shown that the subjects generally preferred the new MUC notation to the original notation. The subjects indicated that the main reason for their preference of the NN was the use of color-based improvement. There were very few questions asked by the subjects during the experiment and in general there were no obvious problems observed during the experiment.

Any future work based on this study can focus on two different areas. The first area involves continuing with MUCM and improving the notation by following the same path taken to improve the MUC in this work. The second area to focus is on the link between MUC and UML. This work largely ignores the link between UML and MUC; however better integration mechanisms are needed to link MUC diagrams with other diagrams in UML.

References

[1] G. Sindre and A. L. Opdahl, “Eliciting security requirements with misuse cases,” Requir. Eng., vol. 10, no. 1, pp. 34–44, 2005.

[2] P. Karpati, G. Sindre, and A. L. Opdahl, “Visualizing cyber attacks with misuse case maps,” in Requirements Engineering: Foundation for Software Quality, Springer, 2010, pp. 262–275.

[3] OMG, “UML 2.4.1,” 2011. [Online]. Available: http://www.omg.org/spec/UML/2.4.1/.

[4] R. J. Buhr, “Use case maps as architectural entities for complex systems,” Softw. Eng. IEEE Trans., vol. 24, no. 12, pp. 1131–1155, 1998.

[5] D. L. Moody, P. Heymans, and R. Matulevi\vcius, “Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation,” Requir. Eng., vol. 15, no. 2, pp. 141–175, 2010.

[6] D. Moody, “The ‘physics’ of notations: toward a scientific basis for constructing visual notations in software engineering,” Softw. Eng. IEEE Trans., vol. 35, no. 6, pp. 756–779, 2009.

[7] J. Lee, “Design rationale systems: understanding the issues,” IEEE Expert, vol. 12, no. 3, pp. 78–85, 1997.

[8] S. Hitchman, “The Details of Conceptual Modelling Notations are Important-A Comparison of Relationship Normative Language,” Commun. Assoc. Inf. Syst., vol. 9, no. 1, p. 10, 2002.

[9] D. Moody and J. van Hillegersberg, “Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML family of diagrams,” in Software Language Engineering, Springer, 2009, pp. 16–34.

[10] OMG, “BPMN 2.0,” 2011. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/.

[11] N. Genon, P. Heymans, and D. Amyot, “Analysing the cognitive effectiveness of the BPMN 2.0 visual notation,” in Software Language Engineering, Springer, 2011, pp. 377–396.

[12] N. Genon, D. Amyot, and P. Heymans, “Analysing the cognitive effectiveness of the UCM visual notation,” in System Analysis and Modeling: About Models, Springer, 2011, pp. 221–240.

[13] P. Mäder and J. Cleland-Huang, “A visual traceability modeling language,” in Model Driven Engineering Languages and Systems, Springer, 2010, pp. 226–240.

[14] C. Shannon and W. Weaver, “Mathematical Theory of Communication,” 1963.

[15] J. Bertin, “Semiology of graphics: diagrams, networks, maps,” 1983.

[16] A. Paivio, Mental representations: A dual coding approach. Oxford University Press, 1990.

[17] D. M. Green, J. A. Swets, and others, Signal detection theory and psychophysics, vol. 1. Wiley New York, 1966.

[18] G. A. Miller, “The magical number seven, plus or minus two: some limits on our capacity for processing information.,” Psychol. Rev., vol. 63, no. 2, p. 81, 1956.

[19] G. Sindre and A. L. Opdahl, “Templates for misuse case description,” in Proceedings of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality (REFSQ’2001), Switzerland, 2001.

[20] G. Sindre and A. Opdahl, “Eliciting security requirements by misuse cases,” in Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Proceedings. 37th International Conference on, 2000, pp. 120–131.

[21] G. Sindre, A. L. Opdahl, and G. F. Brevik, “Generalization/specialization as a structuring mechanism for misuse cases,” in Proceedings of the 2nd symposium on requirements engineering for information security (SREIS’02), Raleigh, North Carolina, 2002.

[22] L. Røstad, “An extended misuse case notation: Including vulnerabilities and the insider threat,” in XII Working Conference on Requirements Engineering: Foundation for Software Quality, Luxembourg, 2006.

[23] I. Biederman, “Recognition-by-components: a theory of human image understanding.,” Psychol. Rev., vol. 94, no. 2, p. 115, 1987.

[24] J. Martin and C. Finkelstein, Information engineering. Savant Research Studies, 1981.

[25] M. Scaife and Y. Rogers, “External cognition: how do graphical representations work?,” Int. J. Hum.-Comput. Stud., vol. 45, no. 2, pp. 185–213, 1996.

[26] M. El-Attar, “From misuse cases to mal-activity diagrams: bridging the gap between functional security analysis and design,” Softw. Syst. Model., vol. 13, no. 1, pp. 173–190, 2014.

[27] G. Sindre, “Mal-activity diagrams for capturing attacks on business processes,” in Requirements Engineering: Foundation for Software Quality, Springer, 2007, pp. 355–366.

[28] G. J. Gorn, A. Chattopadhyay, T. Yi, and D. W. Dahl, “Effects of color as an executional cue in advertising: they’re in the shade,” Manag. Sci., vol. 43, no. 10, pp. 1387–1400, 1997.

[29] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, “Experimentation in Software Engineering: An Introduction,” Kluwer Int. Ser. Softw. Eng., 2000.

[30] M. O. Lehtonen, F. Michahelles, and E. Fleisch, “Trust and security in RFID-based product authentication systems,” Syst. J. IEEE, vol. 1, no. 2, pp. 129–144, 2007.

[31] G. Regev, I. F. Alexander, and A. Wegmann, “Modelling the regulative role of business processes with use and misuse cases,” Bus. Process Manag. J., vol. 11, no. 6, pp. 695–708, 2005.

[32] H. C. Purchase, D. Carrington, and J.-A. Allder, “Empirical evaluation of aesthetics-based graph layout,” Empir. Softw. Eng., vol. 7, no. 3, pp. 233–255, 2002.

[33] S. Siegel and N. Castellan, “Non Parametric Statistics for the Behavioral Sciences,” 1988.

[34] N. Cliff, “Dominance statistics: Ordinal analyses to answer ordinal questions.,” Psychol. Bull., vol. 114, no. 3, p. 494, 1993.

[35] N. Cliff, “Answering ordinal questions with ordinal data using ordinal statistics,” Multivar. Behav. Res., vol. 31, no. 3, pp. 331–350, 1996.

[36] N. Cliff, Ordinal methods for behavioral data analysis. Psychology Press, 1996.

[37] E. Lehmann and H. D’Abrera, Nonparametrics: Statistical methods based on ranks. Prentice Hall (Upper Saddle River, NJ), 1998.

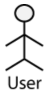
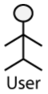
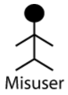
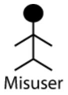
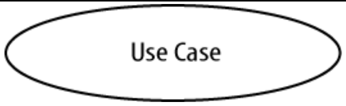
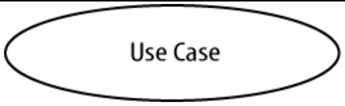
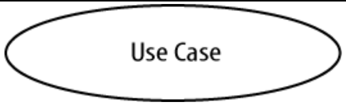
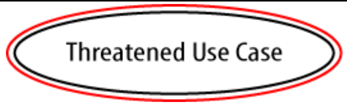
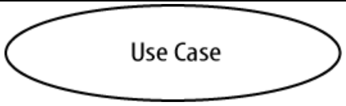
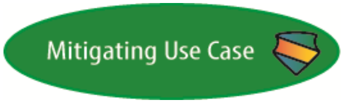
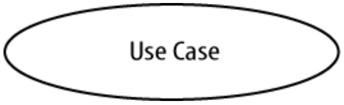
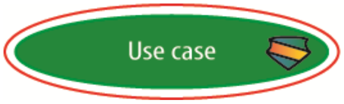
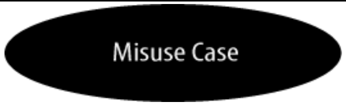
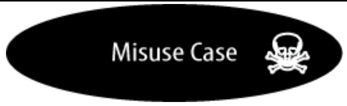
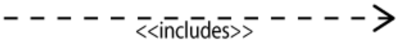
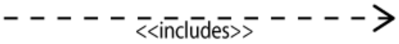
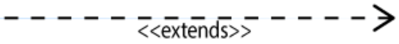
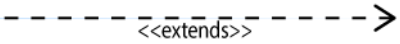
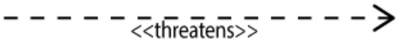
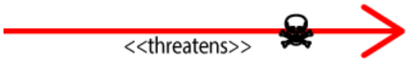
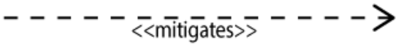
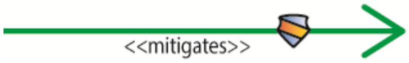
[38] J. D. Kromrey and K. Y. Hogarty, “Analysis options for testing group differences on ordered categorical variables: An empirical investigation of type I error control

and statistical power," Mult. Linear Regres. Viewpoints, vol. 25, no. 1, pp. 70–82, 1998.

[39] J. D. Kromrey, K. Y. Hogarty, J. M. Ferron, C. V. Hines, M. R. Hess, and others, "Robustness in meta-analysis: An empirical comparison of point and interval estimates of standardized mean differences and Cliff's delta," in American Statistical Association 2005 Joint Statistical Meetings, 2005, p. 7.

Appendix A – MUC Study Diagrams

Table 11 Controlled experiment notation legend

Construct	Original Symbol	Proposed Symbol
User		
Misuser		
use case		
Threatened use case		
Mitigating use case		
Threatened and Mitigating use case		
Misuse case		
Includes relation		
Extends relation		
Threatens relation		
Mitigates relation		

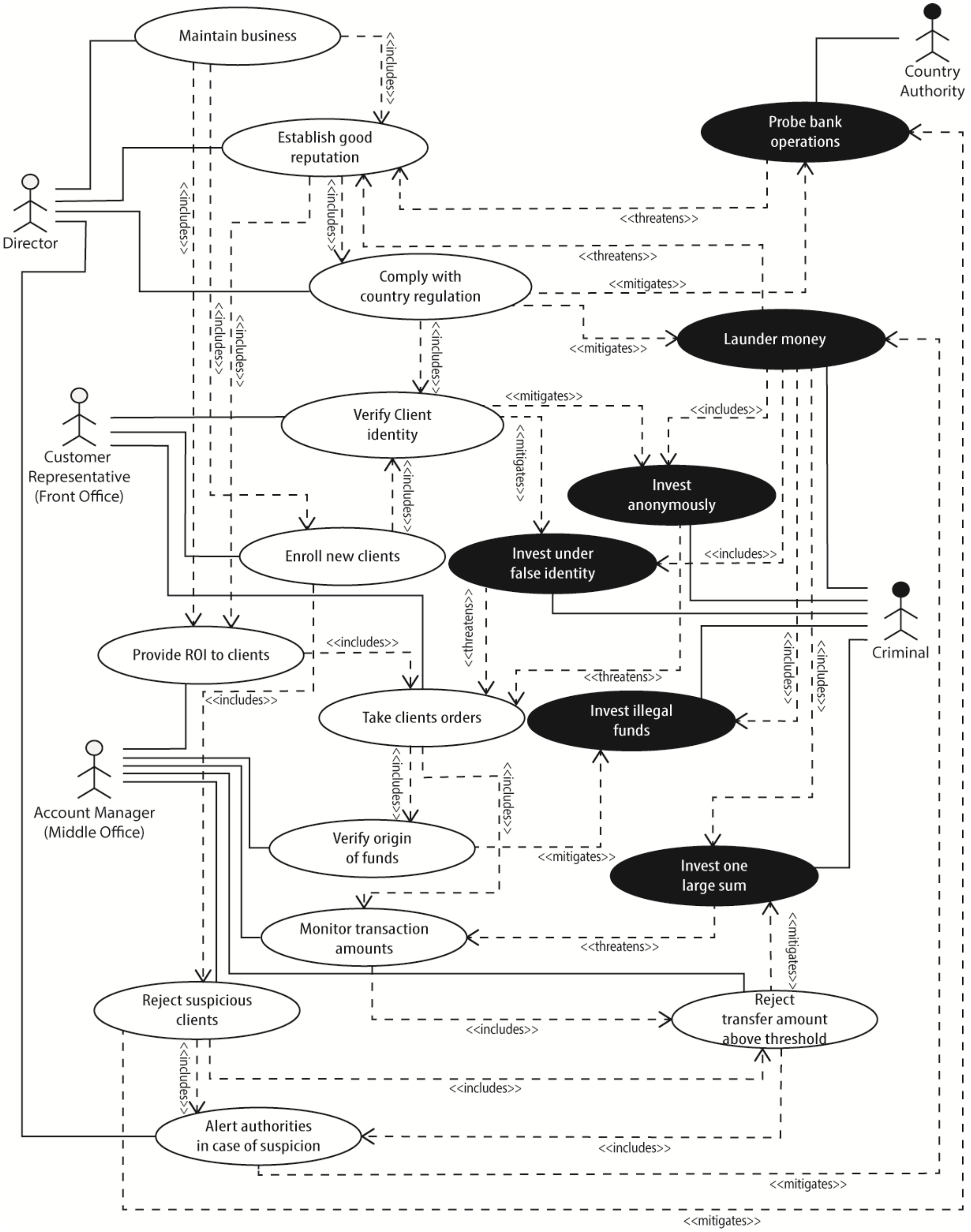


Figure 12 Banking Diagram (Old Notation)

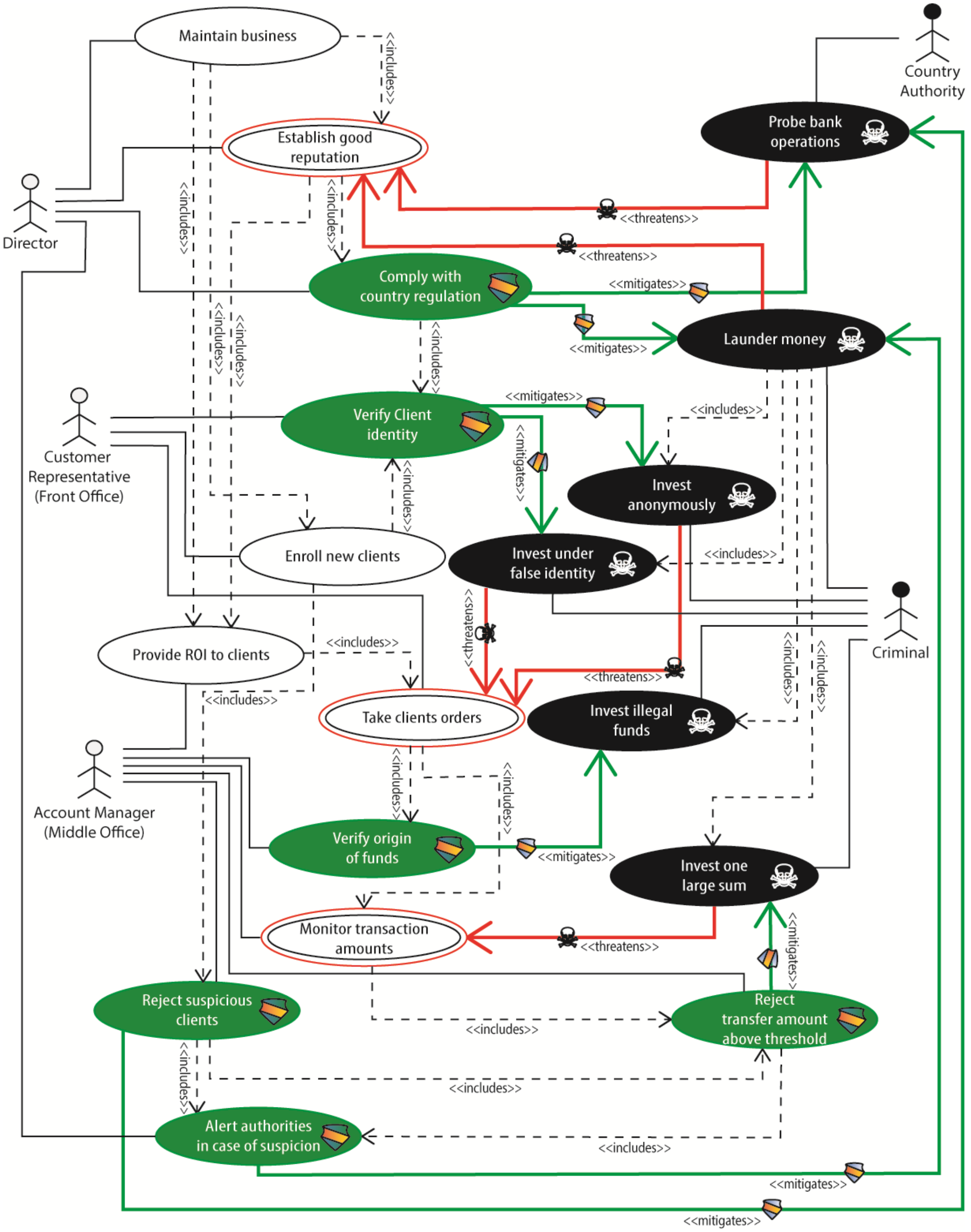


Figure 13 Banking Diagram (New Notation)

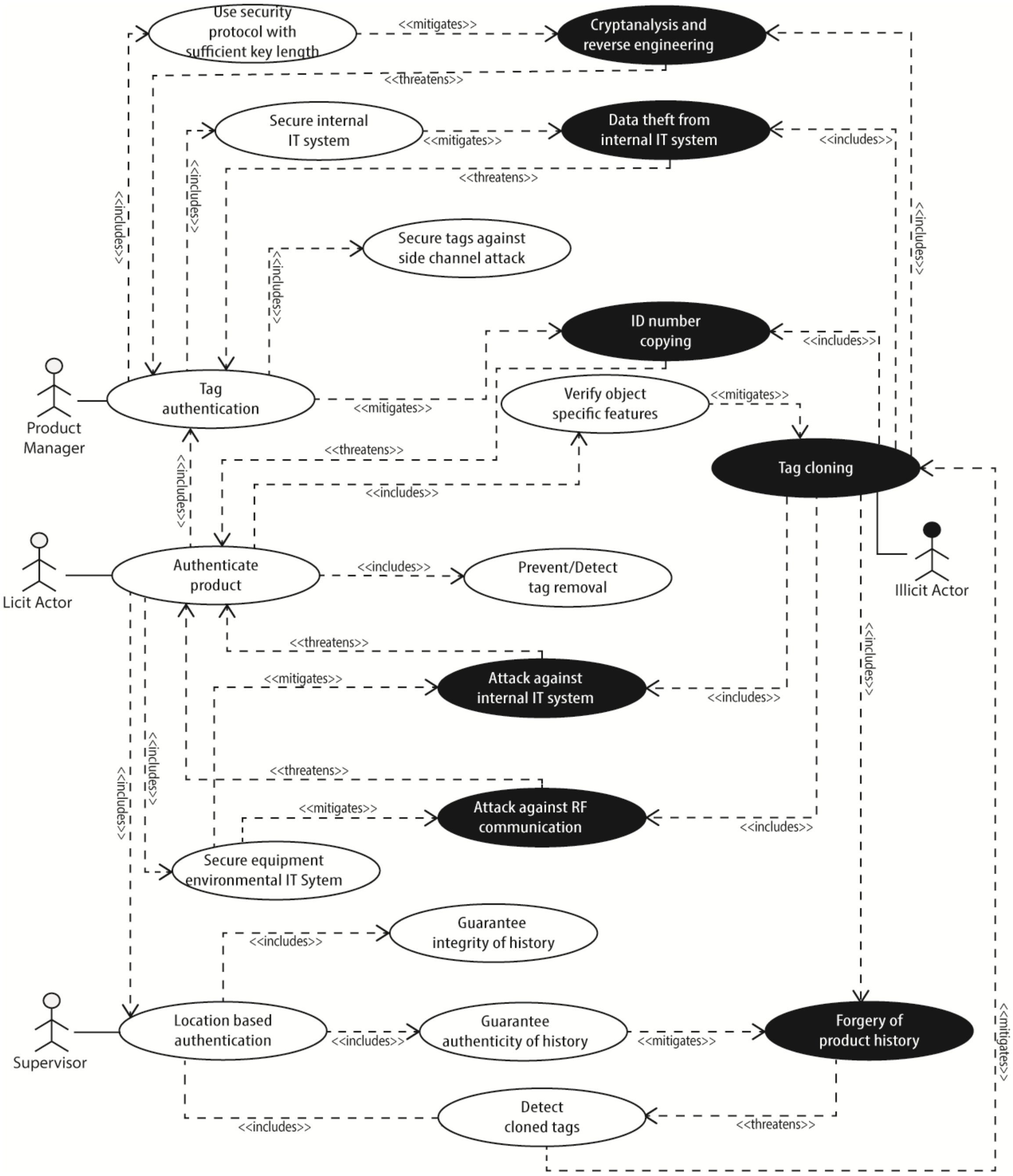


Figure 14 RFID Diagram (Old Notation)

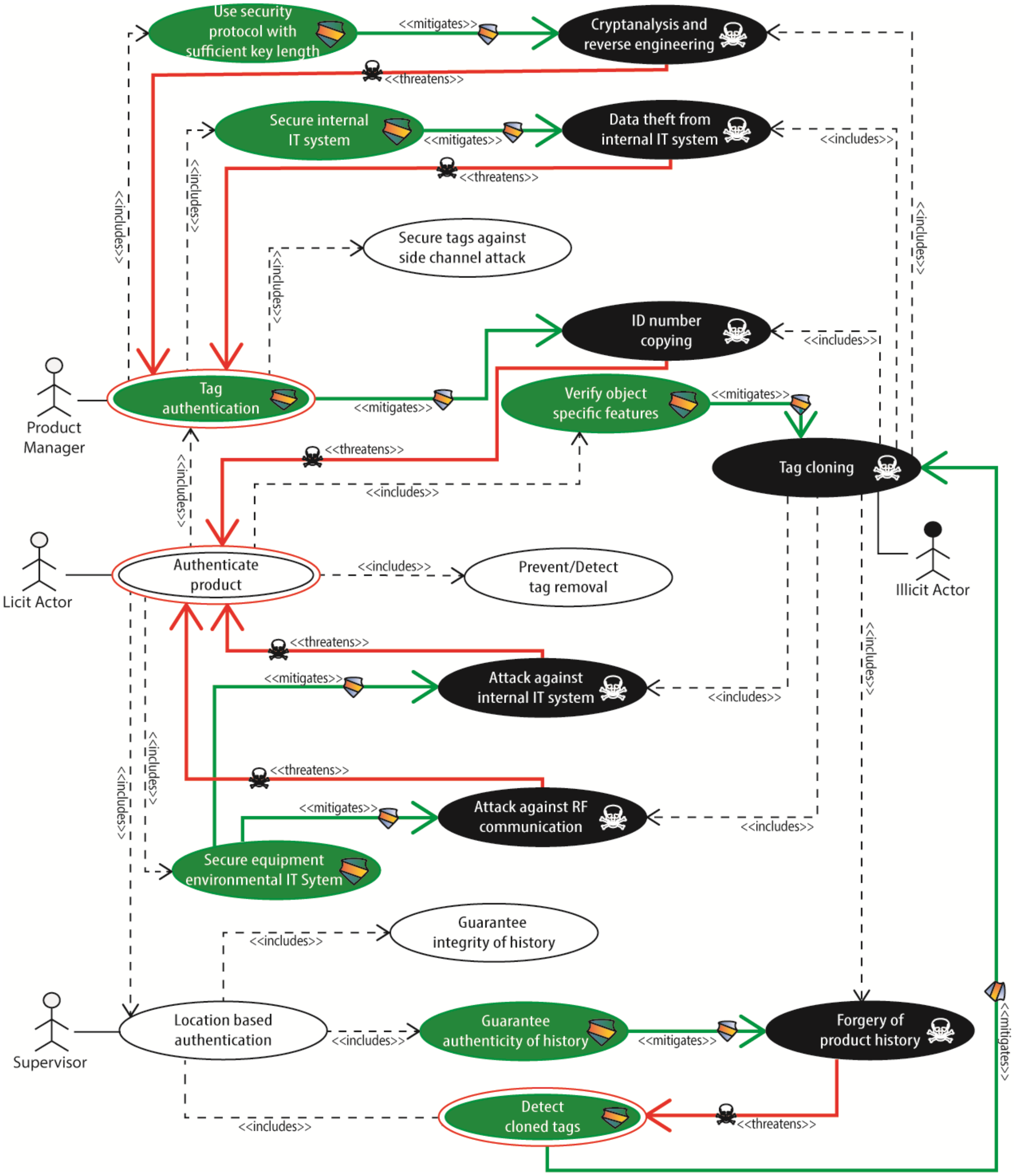

Use security protocol with sufficient key length
<<mitigates>>
Cryptanalysis and reverse engineering
<<threatens>>
Secure internal IT system
<<mitigates>>
Data theft from internal IT system
<<threatens>>
<<includes>>
Secure tags against side channel attack
ID number copying
Product Manager
Tag authentication
<<mitigates>>
Verify object specific features
<<mitigates>>
Tag cloning
<<threatens>>
<<includes>>
Licit Actor
Authenticate product
Prevent/Detect tag removal
Illicit Actor
<<threatens>>
<<mitigates>>
Attack against internal IT system
<<threatens>>
<<mitigates>>
Attack against RF communication
Secure equipment environmental IT Sytem
Guarantee integrity of history
Supervisor
Location based authentication
Guarantee authenticity of history
<<mitigates>>
Forgery of product history
Detect cloned tags
<<threatens>>
<<includes>>

Figure 15 RFID Diagram (New Notation)

Appendix B – MUC Study Questionnaire

Diagram 1 Questionnaire

1. Identify all **threatened** use cases.

- □ Attack against internal IT system
- □ Attack against RF communication
- □ Authenticate product
- □ Cryptanalysis and reverse engineering
- □ Data theft from internal IT system
- □ Detect cloned tags
- □ Forgery of product history
- □ Guarantee authenticity of history
- □ Guarantee integrity of history
- □ ID number copying
- □ Location based authentication
- □ Prevent/detect tag removal
- □ Secure equipment environmental IT system
- □ Secure internal IT system
- □ Secure tags against side channel attack
- □ Tag authentication
- □ Tag cloning
- □ Use security protocol with sufficient key length
- □ Verify object specific features

2. Identify all **mitigating** use cases.

- □ Attack against internal IT system
- □ Attack against RF communication
- □ Authenticate product
- □ Cryptanalysis and reverse engineering
- □ Data theft from internal IT system
- □ Detect cloned tags
- □ Forgery of product history
- □ Guarantee authenticity of history
- □ Guarantee integrity of history
- □ Location based authentication
- □ Prevent/detect tag removal
- □ Secure equipment environmental IT system
- □ Secure internal IT system
- □ Secure tags against side channel attack
- □ Tag authentication
- □ Tag cloning
- □ Use security protocol with sufficient key length
- □ Verify object specific features

- □ ID number copying

3. Identify all use cases that are **threatened AND mitigating**.

- □ Attack against internal IT system
- □ Attack against RF communication
- □ Authenticate product
- □ Cryptanalysis and reverse engineering
- □ Data theft from internal IT system
- □ Detect cloned tags
- □ Forgery of product history
- □ Guarantee authenticity of history
- □ Guarantee integrity of history
- □ ID number copying
- □ Location based authentication
- □ Prevent/detect tag removal
- □ Secure equipment environmental IT system
- □ Secure internal IT system
- □ Secure tags against side channel attack
- □ Tag authentication
- □ Tag cloning
- □ Use security protocol with sufficient key length
- □ Verify object specific features

4. Identify all use cases that are **NOT threatened AND NOT mitigating**.

- □ Attack against internal IT system
- □ Attack against RF communication
- □ Authenticate product
- □ Cryptanalysis and reverse engineering
- □ Data theft from internal IT system
- □ Detect cloned tags
- □ Forgery of product history
- □ Guarantee authenticity of history
- □ Guarantee integrity of history
- □ ID number copying
- □ Location based authentication
- □ Prevent/detect tag removal
- □ Secure equipment environmental IT system
- □ Secure internal IT system
- □ Secure tags against side channel attack
- □ Tag authentication
- □ Tag cloning
- □ Use security protocol with sufficient key length
- □ Verify object specific features

5. Identify all misuse cases with **mitigated relation**.

□ Attack against internal IT system

□ Attack against RF communication

□ Authenticate product

□ Cryptanalysis and reverse engineering

□ Data theft from internal IT system

□ Detect cloned tags

□ Forgery of product history

□ Guarantee authenticity of history

□ Guarantee integrity of history

□ ID number copying

□ Location based authentication

□ Prevent/detect tag removal

□ Secure equipment environmental IT system

□ Secure internal IT system

□ Secure tags against side channel attack

□ Tag authentication

□ Tag cloning

□ Use security protocol with sufficient key length

□ Verify object specific features

6. Identify all misuse cases with **threatens relation**.

□ Attack against internal IT system

□ Attack against RF communication

□ Authenticate product

□ Cryptanalysis and reverse engineering

□ Data theft from internal IT system

□ Detect cloned tags

□ Forgery of product history

□ Guarantee authenticity of history

□ Guarantee integrity of history

□ ID number copying

□ Location based authentication

□ Prevent/detect tag removal

□ Secure equipment environmental IT system

□ Secure internal IT system

□ Secure tags against side channel attack

□ Tag authentication

□ Tag cloning

□ Use security protocol with sufficient key length

□ Verify object specific features

7. Identify the following relations as **E** (Includes/Extends), **T** (Threatens), or **M** (Mitigates).

	Source		**Destination**
___	Attack against internal IT system	→	Authenticate product
___	Attack against RF communication	→	Authenticate product
___	Authenticate product	→	Tag authentication
___	Authenticate product	→	Verify object specific features
___	Authenticate product	→	Prevent/detect tag removal
___	Authenticate product	→	Secure equipment environmental IT system
___	Authenticate product	→	Location based authentication
___	Cryptanalysis and reverse engineering	→	Tag authentication
___	Data theft from internal IT system	→	Tag authentication
___	Detect cloned tags	→	Tag cloning
___	Forgery of product history	→	Detect cloned tags
___	Guarantee authenticity of history	→	Forgery of product history
___	ID number copying	→	Authenticate product
___	Location based authentication	→	Guarantee integrity of history
___	Location based authentication	→	Guarantee authenticity of history
___	Location based authentication	→	Detect cloned tags
___	Secure equipment environmental IT system	→	Attack against internal IT system
___	Secure equipment environmental IT system	→	Attack against RF communication
___	Secure internal IT system	→	Data theft from internal IT system
___	Tag authentication	→	Use security protocol with sufficient key length
___	Tag authentication	→	Secure tags against side channel attack
___	Tag authentication	→	Secure internal IT system
___	Tag authentication	→	ID number copying
___	Tag cloning	→	Cryptanalysis and reverse engineering

___ Tag cloning → Attack against internal IT system

___ Tag cloning → Attack against RF communication

___ Tag cloning → Forgery of product history

___ Tag cloning → Data theft from internal IT system

___ Tag cloning → ID number copying

___ Use security protocol with sufficient key length → Cryptanalysis and reverse engineering

___ Verify object specific features → Tag cloning

Diagram 2 Questionnaire

1. Identify all **threatened** use cases.

□ Alert authorities in case of suspicion

□ Comply with country regulation

□ Enroll new clients

□ Establish good reputation

□ Invest anonymously

□ Invest illegal funds

□ Invest one large sum

□ Invest under false identity

□ Launder money

□ Maintain business

□ Monitor transaction amounts

□ Probe bank operations

□ Provide ROI to clients

□ Reject suspicious clients

□ Reject transfer amount above threshold

□ Take client orders

□ Verify client identity

□ Verify origin of funds

2. Identify all **mitigating** use cases.

□ Alert authorities in case of suspicion

□ Comply with country regulation

□ Enroll new clients

□ Establish good reputation

□ Invest anonymously

□ Invest illegal funds

□ Invest one large sum

□ Invest under false identity

□ Launder money

□ Maintain business

□ Monitor transaction amounts

□ Probe bank operations

□ Provide ROI to clients

□ Reject suspicious clients

□ Reject transfer amount above threshold

□ Take client orders

□ Verify client identity

□ Verify origin of funds

3. Identify all use cases that are **threatened AND mitigating**.

□ Alert authorities in case of suspicion

□ Comply with country regulation

□ Enroll new clients

□ Establish good reputation

□ Invest anonymously

□ Invest illegal funds

□ Invest one large sum

□ Invest under false identity

□ Launder money

□ Maintain business

□ Monitor transaction amounts

□ Probe bank operations

□ Provide ROI to clients

□ Reject suspicious clients

□ Reject transfer amount above threshold

□ Take client orders

□ Verify client identity

□ Verify origin of funds

4. Identify all use cases that are **NOT threatened AND NOT mitigating**.

□ Alert authorities in case of suspicion

□ Comply with country regulation

□ Enroll new clients

□ Establish good reputation

□ Invest anonymously

□ Invest illegal funds

□ Invest one large sum

□ Invest under false identity

□ Launder money

□ Maintain business

□ Monitor transaction amounts

□ Probe bank operations

□ Provide ROI to clients

□ Reject suspicious clients

□ Reject transfer amount above threshold

□ Take client orders

□ Verify client identity

□ Verify origin of funds

5. Identify all misuse cases with **mitigated relation**.

□ Alert authorities in case of suspicion

□ Comply with country regulation

□ Enroll new clients

□ Establish good reputation

□ Invest anonymously

□ Invest illegal funds

□ Invest one large sum

□ Invest under false identity

□ Launder money

□ Maintain business

□ Monitor transaction amounts

□ Probe bank operations

□ Provide ROI to clients

□ Reject suspicious clients

□ Reject transfer amount above threshold

□ Take client orders

□ Verify client identity

□ Verify origin of funds

6. Identify all misuse cases with **threatens relation**.

□ Alert authorities in case of suspicion

□ Comply with country regulation

□ Enroll new clients

□ Establish good reputation

□ Invest anonymously

□ Invest illegal funds

□ Invest one large sum

□ Invest under false identity

□ Launder money

□ Maintain business

□ Monitor transaction amounts

□ Probe bank operations

□ Provide ROI to clients

□ Reject suspicious clients

□ Reject transfer amount above threshold

□ Take client orders

□ Verify client identity

□ Verify origin of funds

7. Identify the following relations as **E** (Includes/Extends), **T** (Threatens), or **M** (Mitigates).

	Source		Destination
___	Alert authorities in case of suspicion	→	Launder money
___	Comply with country regulation	→	Probe bank operations
___	Comply with country regulation	→	Launder money
___	Comply with country regulation	→	Verify client identity
___	Enroll new clients	→	Verify client identity
___	Enroll new clients	→	Reject suspicious clients
___	Establish good reputation	→	Comply with country regulation
___	Establish good reputation	→	Provide ROI to clients
___	Invest anonymously	→	Take client orders
___	Invest one large sum	→	Monitor transaction amounts
___	Invest under false identity	→	Take client orders
___	Launder money	→	Establish good reputation
___	Launder money	→	Invest anonymously
___	Launder money	→	Invest under false identity
___	Launder money	→	Invest illegal funds
___	Launder money	→	Invest one large sum
___	Maintain business	→	Establish good reputation
___	Maintain business	→	Enroll new clients
___	Maintain business	→	Provide ROI to clients
___	Monitor transaction amounts	→	Reject transfer amount above threshold
___	Probe bank operations	→	Establish good reputation
___	Provide ROI to clients	→	Take client orders
___	Reject suspicious clients	→	Probe bank operations
___	Reject suspicious clients	→	Reject transfer amount above threshold
___	Reject suspicious clients	→	Alert authorities in case of suspicion

___ Reject transfer amount above threshold → Invest one large sum

___ Reject transfer amount above threshold → Alert authorities in case of suspicion

___ Take client orders → Verify origin of funds

___ Take client orders → Monitor transaction amounts

___ Verify client identity → Invest anonymously

___ Verify client identity → Invest under false identity

___ Verify origin of funds → Invest illegal funds

<u>Questions</u>

Q1- Which notation did you like the most (New Notation or Old Notation)?

Q2- Why did you like the notation you chose? What was your experience when performing the exercises?

Q3- How do you think the notation you chose can be improved?

Appendix C – MUC Study Raw Results

Table 12 Response times for Group A and Group B

Group A Time (Seconds)			***Group B Time (Seconds)***		
Subject	***Diagram 1 (ON)***	***Diagram 2 (NN)***	***Subject***	***Diagram 1 (NN)***	***Diagram 2 (ON)***
A1	1189	644	B1	571	825
A2	867	500	B2	727	1282
A3	1397	683	B3	1387	1452
A4	954	450	B4	1001	1119
A5	857	540	B5	945	1334
A6	1590	743	B6	753	972
A7	1048	531	B7	472	839
A8	905	584	B8	528	780
A9	763	482	B9	663	785
A10	870	485	B10	724	843
A11	1226	500	B11	736	867
A12	780	630	B12	650	734
A13	1023	556	B13	778	1145
A14	1444	728	B14	555	777
A15	642	482	B15	686	1040
A16	1130	550	B16	720	1020
A17	934	759	B17	1800	2400

Table 13 Errors committed for Group A

Group A Errors						
	Diagram 1			***Diagram 2***		
Subject	***Wrong Identification***	***Overlooked***	***Total***	***Wrong Identification***	***Overlooked***	***Total***
A1	3	3	6	0	0	0
A2	2	0	2	0	1	1
A3	11	0	11	4	5	9
A4	0	0	0	6	0	6
A5	0	1	1	0	0	0
A6	1	2	3	1	0	1
A7	2	1	3	1	0	1
A8	14	1	15	0	0	0
A9	0	13	13	10	0	10
A10	0	2	2	0	9	9
A11	0	1	1	0	0	0
A12	0	2	2	0	0	0
A13	6	10	16	1	1	2
A14	11	1	12	0	0	0
A15	1	4	5	17	0	17
A16	0	0	0	5	8	13
A17	1	1	2	0	0	0

Table 14 Errors committed for Group B

Group B Errors						
Diagram 1				***Diagram 2***		
Subject	***Wrong Identification***	***Overlooked***	***Total***	***Wrong Identification***	***Overlooked***	***Total***
B1	0	0	0	0	3	3
B2	0	0	0	0	0	0
B3	1	0	1	0	0	0
B4	1	0	1	0	0	0
B5	7	0	7	5	3	8
B6	0	0	0	0	0	0
B7	0	0	0	3	6	9
B8	13	0	13	6	7	13
B9	0	0	0	1	2	3
B10	0	0	0	0	0	0
B11	0	0	0	1	1	2
B12	0	0	0	0	1	1
B13	4	0	4	0	0	0
B14	0	0	0	1	1	2
B15	8	0	8	0	0	0
B16	1	0	1	0	4	4
B17	0	1	1	0	1	1

Appendix D – MUC Study Raw Qualitative Data

1- Which notation did you like the most (ON or NN)?
2- Why did you like the about the notation you chose? What was your experience when performing the exercises?
3- How do you think the notation you chose can be improved?

Table 15 Qualitative results for Group A

Group A Qualitative Results			
Subject	***Q1***	***Q2***	***Q3***
A1	NN	Filling the diagram with color makes it clearer and quicker at identifying the kind of use case	Satisfied and add nothing
A2	NN	it is easier to get	No Answer
A3	Neither	No Answer	Add symbols/colors for include and extends.
A4	No Answer	No Answer	No Answer
A5	NN	Sight goes to it directly. The original diagram we can't see what's wanted properly, while the proposed is much clearer.	Increase size of red border for threatened use case
A6	NN	Easier and faster to read.	NN
A7	NN	Easier to deal with. Easier to detect and infer information is quicker.	Maybe put the first letter of the relationship at the start of arrow in use case. (See provided drawing)
A8	NN	Because it is easier to recognize and less error.	Differentiate between extends and includes
A9	NN	Easier to look at and get information from.	No need for icons, only color will do.
A10	NN	Colorful, felt like no need to double check.	Use icons.
A11	NN	Easy to understand, easy to follow relation arrows and attractive.	No idea

A12	NN	Simple and easy to understand.	Maybe having symbols for includes and extends.
A13	NN	Easier to catch (Find), simple	The icon used can be more simpler
A14	NN	it grasps the attention and you can know the relation just by looking and it saves time	adding color for extend and includes relations
A15	NN	easier to understand and follow	I think it is perfect
A16	NN	You can tell the kind of use case without tracing, more comfortable.	make different colors for extend and includes
A17	NN	It is easier, the original notation is confusing as you have to identify the relation in order to id the use case, while in the proposed we can tell just by looking at the use case	Differentiate between the threatening and mitigated misuse cases, just like the use cases.

Table 16 Qualitative results for Group B

Group B Qualitative Results			
Subject	***Q1***	***Q2***	***Q3***
B1	NN	Easy to notice and gives better understanding with less amount of thinking. Don't have to concentrate as much.	Improve the includes/extends relations. Bold for includes.
B2	NN	Faster to grasp, less boring and more fun.	No Answer
B3	NN	Easy to understand.	perfect
B4	No Answer	No Answer	No Answer
B5	NN	speedy and easy to understand and find	Differentiate between include/extends and color the normal use case orange.
B6	NN	Colorful and easy to track use cases and their relations.	differentiate between include/extends
B7	NN	Easier and less time consuming.	• Color printing is costly. • Using smaller icons near use case will be useful. • Have different style of lines connecting use cases.
B8	NN	Colored diagram was easy while the black and white caused a headache.	The lines connections could be dotted or thick.
B9	NN	It is more cleaner, you can immediately see all the relations and understand diagram quickly. It is much more fun to read and extract information from.	There will be always room before improvements, but the idea is great. Hope this will be use case standard.
B10	NN	It is clear, I felt comfortable when reading the one with proposed notation and confident about my answers.	Nothing.
B11	ON	In the original, I just check the use case and the relation between them. The proposed notation, I check the use case and also I check its shape and I need to remember the shapes.	minimize the number of shapes

B12	NN	It is easier to grasp and understand.	if there was a way to arrange the use cases and the actors and relations links
B13	NN	It is simple.	No Answer
B14	NN	It is easier to read and understand.	No Answer
B15	No Answer	No Answer	No Answer
B16	NN	Very easy to follow and comprehend	No Answer
B17	No Answer	No Answer	No Answer

Vitae

Name : Faisal Saleh

Nationality : Pakistani

Date of Birth : 20/9/1986

Email : faisal86@me.com

Address : L.Y. 11/36, 5th Floor, Mashallah Manzil, Moosa Lane, Karachi, Pakistan.

Academic Background : Bachelor of Science in Computer Science, Institute of Business Administration, Karachi, Pakistan.

Work Experience : *September 2009 – 2012*: *Senior Software Engineer*. Creative Chaos (Pvt.) Limited, Karachi, Pakistan

June 2008 – May 2009: *Software Engineer*.Wallsoft eSolutions Platform, Karachi, Pakistan

Published Papers : Issam Laradji, Faisal Saleh, Musab A. AlTurki. *"Sparse Single-hidden Layer Feedforward Neural Network for Semantic Parsing",* 24th International Conference on Artificial Neural Networks

Mazin Saeed, Faisal Saleh, Sadiq Al-Insaif and Mohamed El-Attar. "Evaluating the cognitive effectiveness of the visual syntax of feature diagrams", Asia Pacific Requirements Engineering Symposium