EVALUATION OF THE COGNITIVE EFFECTIVENESS FOR THE VISUAL SYNTAX OF FEATURE DIAGRAMS AND SECURE TROPOS

BY

Mazin Ali Saeed

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 **Mazin Ali Saeed** under the direction 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)

Dr. Adel Ahmed

Department Chairman

Dr. Mohammad Alshayeb
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Mahmoud O. Elish

(Member)

12/6/14

Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

I would like to dedicate my thesis work to my beloved family. A special feeling of gratitude to my parents and sister for their encouragements, care and love. I would like also to thank my brother *Mohamed Ali* for his continues support throughout pursuing my master degree. This thesis won't be accomplished without their presence.

I would like also to thank Dr Mohamed Saeed and Dr Saleh Duffua for their guidance and motivations. Finally I would like to thank friends whom showed support and backed me up.

ACKNOWLEDGMENTS

First of all I would like to thank almighty Allah for giving me the ability and determination to complete my master degree.

I would like to present my sincere gratitude to my advisor Dr. Mohamed El-Attar for his guidance and support he provided throughout this research. I feel very fortunate to have had the opportunity to work under his supervision. Dr El-Attar devoted his time and energy to help me complete this research, he had been welcoming person, every walk to his office was pleasurable, I learned a lot with his accompany, I no longer see Dr El-Attar as a teacher but as brother and a friend.

I would also like to thank the committee members, Dr. Mohammad Alshayeb and Dr. Mahmoud O. Elish, for dedicating time out of their busy schedule for this work and providing their feedback and comments. I also would like to thank other professors and graduate students who participated by comments and feedback.

Lastly, I would like to thank Faisal, we shared nice time working together during last two semesters. Certainly, I will miss countless “building 24” breaks that were undertaken during our work on the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUML	:	Agent Unified Modeling Language.
ERP	:	Enterprise Recourse Planning
FD	:	Feature Diagram.
NN	:	New Notation.
ON	:	Old Notation.
PoN	:	Physics of Notation.
RE	:	Requirement Engineering.
SE	:	Software Engineering.
SPL	:	Software Product Line.
ST	:	Secure Tropos.
UML	:	Unified Modeling Language

ABSTRACT

Full Name : Mazin Ali Ahmed Saeed

Thesis Title : Evaluation of the Cognitive Effectiveness for the Visual Syntax of Feature Diagrams and Secure Tropos

Major Field : Computer Science

Date of Degree : May, 2014

[Context and Motivation] Models are widely used in Software Engineering to capture and communicate different development artifacts. Feature models contain feature diagrams that graphically depict features in a hierarchical form. Secure Tropos, which is an instance of i* framework adopting security constructs to show a security goal orient diagram. [Problem/Question] Many research works have been devoted to enriching the visual syntax several diagrams to extend its expressiveness to capture additional types of semantics; however, there is a lack of research that evaluating the visual perception of notations by its readers. Models serve a dual purpose: to brainstorm and communicate. A very sophisticated yet unreadable model is arguably useless. To date, there has not been a scientific evaluation of the cognitive effectiveness of the visual syntax of feature diagrams and secure tropos. [Principle Ideas] This study presents a scientific evaluation of the cognitive effectiveness of feature diagrams and secure tropos. The evaluation is based on theory and empirical evidence mainly from the cognitive science field. [Contribution] The evaluation reveals drawbacks in the visual notation of both diagrams. This study also includes some recommendations for improvement to remedy the identified flaws in feature diagram. Empirical study shows improvements in term of time response and in term of mistake committed by participant in the study.

ملخص الرسالة

الاسم الكامل :ما زن علي أحمد سعيد

عنوان الرسالة: تقييم فاعليه الإدراك للتراكيب البصريه في الرسوم البيانيه للميزات و تروبوس الآمنه.

التخصص: علوم الحاسب

تاريخ الدرجة العلمية: مايو – 2014

(المحتوى و الحافزيه) تستعمل النماذج بشكل واسع في هندسه البرمجيات لإلتقاط و توصيل مخرجات التطوير. نماذج المزايا تحتوي على رسوم بيانيه للمزايا و التي تصور الخصائص و المزايا في شكل هرمي. تروبوس الآمنة، و التي هي مثيل لإطار *i، تتبنى البناءات الآمنة لتنتج مخططات الأهداف ذات التوجه الأمني. (المشكله/السؤال) العديد من الابحاث اهتمت لإثراء الجمل المرئيه للعديد من المخططات لاضافه قابليه المخططات على احتواء تعابير جديده، لكن هنالك نقص في الابحاث التي تقيم الاستقبال المرئي لهذه المخططات من قبل القار .. تستعمل النماذج لغرضان: التفكير و التواصل. النماذخ المتمرسه و لكن صعبه القراءه يمكن القول عنها انها غير مفيده. الى هذه اللحظه، لايوجد تقييم علمي لفاعليه ادراك التراكيب البصريه لرسوم المزايا و تروبوس الآمنه. (الفكره الأساسيه) هذه الدراسه تقدم تقييم علمي لفاعليه ادراك رسوم المزايا و تروبوس الآمنه. بني التقييم على نظريات و أدله مجربه بشكل اساسي من مجال علم الإدراك. (المساهمه) تكشف الدراسه عن مساوء في التراكيب البصريه في الرسوم البيانيه للميزات و تروبوس الآمنه. هذه الدراسه ايضا تتضمن اقتراحات لتطوير و علاج التراكيب البصريه لرسوم المزايا. ايضا هذا البحث قام بعمل درسه تظهر ان الاقتراحات قامت بتحسين الرسوم البيانيهمن ناحيه الاستجابه الزمنيه ومن ناحيه الأخطاء المرتكبه من قبل المشاركين في البحث.

CHAPTER 1

INTRODUCTION

Visual notations commonly used to model different aspects of systems during software development. Multiple notations mused to represent the system on different levels of abstractions. These notations form a fundamental part in software engineering practices. The core purpose of models is to convey the different aspects of a system to multiple stakeholders of the system. However, the effectiveness of models in conveying information correctly can vary. If the reader of a model misreads or misinterprets it, then the intrinsic goal of the modeling exercise has failed. A misread or misinterpreted model can lead to the development of end products that do not possess the correct desired characteristics as intended by its stakeholders.

Generally, most researchers devoted their time in enriching the visual syntax of models to extend its expressiveness to capture additional semantics and operations. While such stream of research provides valuable contributions to the field of software engineering, it is arguably equally as important to investigate the visual perception of the visual syntax by its readers. Researchers decisions while designing visual symbols for software engineering notations are made subjectively, without presenting any intuition on the selection process[1]–[3]. These design choices can affect the cognitive expressiveness of the notation as they need to be easily comprehensible by the human mind for precise communication [4] however, there lacks research that evaluate the cognitive effectiveness of visual diagrams. Cognitive effectiveness for software engineering notations is defined as “the speed, ease and accuracy with which a representation can be processed by the human mind” [5]. The most outstanding reason for neglecting this stream of research is the lack of a theoretical basis to conduct notation evaluations scientifically.

In 2009, Moody [1] documented nine principles for evaluating and designing cognitively effective notations in what is considered the seminal paper in the area of cognitive effectiveness evaluation in the software engineering field. The nine principles were compiled and collated from theory and empirical evidence mainly from the cognitive science field, among other fields. The nine principles focus on the visual perception of notations rather than their coverage of semantic constructs. This study will present a scientific evaluation of the cognitive effectiveness of Feature diagrams and Secure Tropos, with an aim to highlight areas where the notation violates the principle and suggest improvements that adhere to the aforementioned principles. Additionally, this study will provide suggestions for improvement to remedy the identified flaws; critical aspect of Feature Models modeling that has so far been overlooked. Importance of the aforesaid diagrams comes from the fact that they are used during early phase of stage of requirements analysis; both diagrams require interaction with non-technical stakeholders, poor communication and poor mutual understanding can severely impact the overall process of producing the desired system.

1.1. Research Questions

Many research works have been devoted to enriching the visual syntax of feature diagrams and secure Tropos to extend its expressiveness to capture additional types of semantics, however, there is a lack of research that evaluate the visual perception of these models by its readers. Models serve a dual purpose: to brainstorm and communicate. A very sophisticated yet unreadable model is arguably useless. To date, there has not been a scientific evaluation of the cognitive effectiveness of the visual syntax of either feature diagrams or secure Tropos. This study focuses on two questions:

- What are the deficiencies in Feature Diagram and Secure Tropos modeling techniques according to Physics of Notation?
- Does resolving the deficiencies highlighted by Physics of Notation actually improve feature diagram?

1.2. Thesis Objectives

The major objective of proposed study is to present a scientific evaluation of the cognitive effectiveness of the aforementioned diagrams. The evaluation approach based on theory and empirical evidence mainly from the cognitive science field. In order to carry out this objective the following tasks will be executed:

1. Evaluate the features model notation using principles presented in “Physics of Notation" and accordantly formulate suggested improvements that fulfill the principles when applicable.

2. Evaluate the secure Tropos notation using principles presented in “Physics of Notation".
3. Establish an empirical evaluation on the proposed notation for feature diagrams that contains the changes suggested in step 1. The evaluation determines whether the effects of the changes proposed are positive or negative impact compared to the original notation.

1.3. Validation

Feature diagram suggested improvements will be validated using questionnaires; such type of analysis will help to obtain meaningful statistical data. Validation will be carried out with the help of two student groups and a group of professionals.

1.4. Contribution

The contribution of the study are extensions and enhancements to feature diagram and secure tropos notations that comply with the cognitive effective principles presented in “Physics of Notations”. Empirical evaluation of feature diagram extension showed better improvements over the original notation in term of time response and error commitments.

1.5. Thesis Outlines

The remainder of this thesis is organized as follows: chapter 2 presents literature review, background on Physics of Notations, feature diagrams and secure Tropos. chapter 3 analysis feature diagram and secure tropos based on PoN, highlighting problems in the notations that effect the cognition effectiveness, and suggested improvement to the feature diagram visual symbols. In chapter 4, an empirical evaluation for the new notation of feature diagram is studied in quantitative and qualitative manner. Finally, Chapter 5 concludes the thesis and presents some directions in which future work can be done. All information used to conduct the empirical evaluation and all the results of the evaluation attached at end as appendix

CHAPTER 2

LITERATURE REVIEW

2.1. Principles for Cognitively Effective Visual Notations (Physics of Notations)

The evaluation of the cognitive effectiveness based on nine evidence-based principles that provide a theoretical basis for designing and evaluating a visual syntax. The nine principles are: Semiotic Clarity, Perceptual Discriminability, Semantic Transparency, Complexity Management, Cognitive Integration, Visual Expressiveness, Dual Coding, Graphic Economy and Cognitive Fit, Figure 1 illustrate the principles. Each principle is defined and briefly summarized in its corresponding subsection.

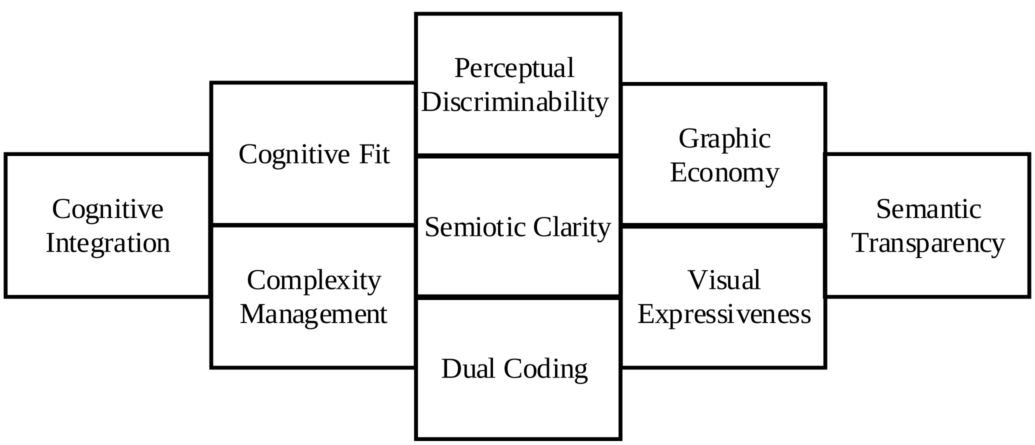


Figure 1 Physics of Notation principles

Two spaces required attention while evaluating the cognitive effectiveness of a visual notation, the problem space and the graphic space. Problem Space gathers all the semantics constructs proposed by its notations. Such semantics founded in a metamodel or a standardized document. Nevertheless, a literature review on the subject of evaluation is useful, due to absent of such formal source. The Graphic Space or the “The Visual Alphabet” is the set of all potential graphical encodings (visual variables). According to [1], [5], [6], there are eight fundamental visual variables that can be used to encode semantics graphically. The eight visual variables are shape, brightness, size, orientation, color, texture, horizontal and vertical positions. Notation designers can use combinations of these variables to create an infinite number of graphical symbols. As a prelude to evaluating the symbols, it is required to determine the visual variables used by each symbol and the values of each visual variable used. The third space that conceptualized by the “Physics of Notations” [1] is the Solution Space. The Solution Space is concerned with choosing the most cognitively effective set of symbols produced out of the endless combinations from the Graphic Space. The Solution Space considered for suggesting improvements to the suboptimal design aspects of a visual language.

2.1.1. Principle of Semiotic Clarity:

Principle of semiotic clarity simply stating there should not be more than one-to-one mapping between semantics constructs and the visual symbols. If there is more than a one-to-one instance between semantics and symbols, one or more of the subsequent anomalies can arise. Figure 2 describes the anomalies and their relations:

Symbol Redundancy (synography): multiple graphical symbols used to represent one semantic.

Symbol Overload (homography): multiple semantics represented by one symbol.

Symbol deficit: there is no symbol exists to represent a particular semantic.

Symbol excess: there is a symbol does not represent any semantic.

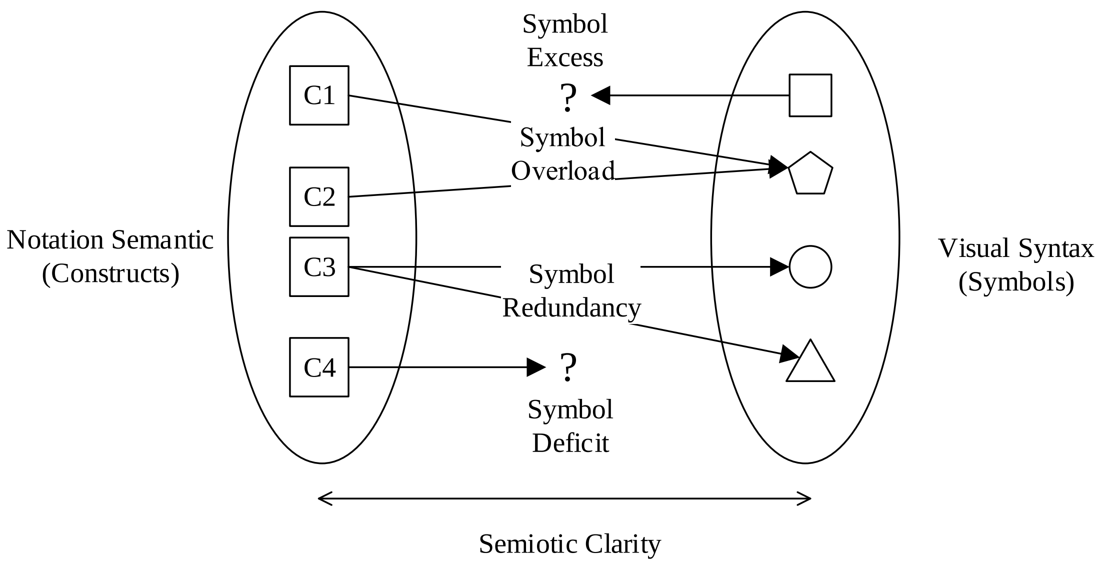


Figure 2 semiotic clarity

- **Symbol Redundancy**

Multiple symbols used to represent one construct. Symbol redundancy carries an encumbrance of choices to the notation modeler and the notation receiver equally. Since the modeler have to decide which symbol to implement correctly, (although there is no basis for judgment) and on the other side the receiver or the reader of the notation have to acknowledge and memorize different variation of the same visual syntax. Therefore, that situation leads to confusion, and consequently, symbol redundancy increases the learning curve required to effectively use the relative diagram. Figure 3 shows a possible instance of redundancy in a visual notation.

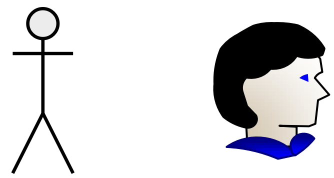


Figure 3 Two different representation of actors.

- **Symbol Overload**

Symbol overload or Homographs (similar to homonyms in linguistics), described in [1] as the worst among anomalies as it leads to uncertainty and probably misunderstanding. It contradict a basic characteristic of symbols notational system that each symbol should represent only one meaning. Symbol overload directly leads to confusion since there is no means to visually determine the semantic that is conveyed by a particular symbol.

- **Symbol Excess & Symbol Deficit**

Symbol excess enlarge diagram complexity that has a direct effect in Complexity Management and Graphic Economy (detailed later), such complexity added to the diagram can negatively impact understanding [1]. Oppositely, symbol deficit facilitate maintaining diagram complexity, it is unfavorable to show all semantics in a diagram.

2.1.2. Visual Expressiveness

The principle of Visual Expressiveness for a notation is concerned with the number of visual variables (Figure 4) used and the ranges of values used in each variable. The larger the number of visual variables a notation uses (information variables degree see Figure 5) and the wider the ranges of values used in each variable, the more visually expressive the notation becomes.

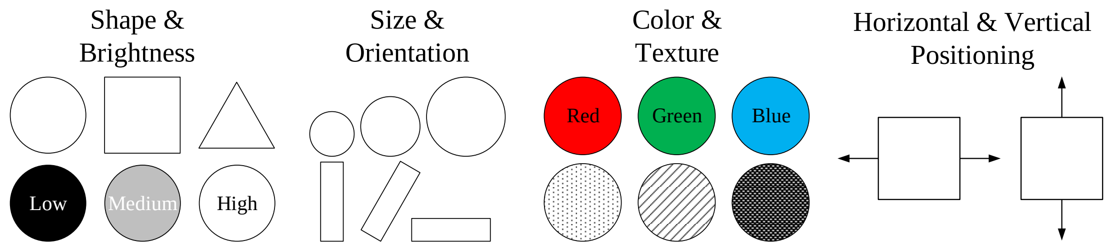


Figure 4 Visual Variables

A notice should be taken to avoid confusion of this principle and the principle of Perceptual Discriminability, which measure the distance between symbols in pairwise**.**

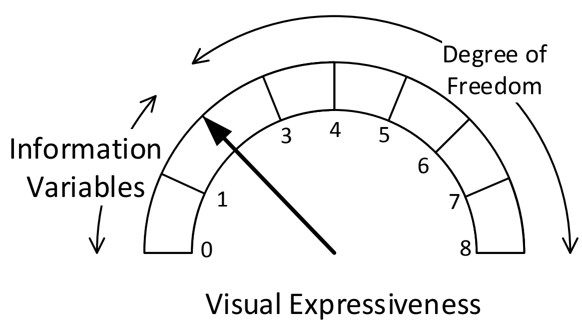


Figure 5 Visual variables degree

2.1.3. Perceptual Discriminability

The principle of Perceptual Discriminability refers to the ease by which symbols from one notational set can be differentiated from each other. This principle requires the language designers to increase the visual distance between symbols within a notational set, by apply the visual variables defined by Bertin et al. [7]. Visual distance, which is the number of visual variables that differ between two symbols is increased by utilizing a large number of visual variables and utilizing a wider range of values for each variable. Larger visual distance between symbols allows them to be easily differentiated from each other, accordingly yields to better perceptual differentiation. This analysis highlights symbols that are similar and thus have low levels of perceptual discriminability. Here similarity is determined by the shape variable since it has the greatest influence on cognition by humans [1].

2.1.4. Complexity Management

The principle of Complexity Management is described as “the ability of a visual notation to represent information without overloading the human mind” [1]. The presence of mechanisms that support the handling of complexity is essential for visual notations. High levels of complexity can severely limit model comprehension especially amongst novices. Complexity can dealt with through modularization and hierarchical structuring. Modularity is a common practice to break down complexity by dividing large components into smaller ones. Figure 6 illustrates as an example, UML Use Case diagram, such complex diagram can affect understandability for users.

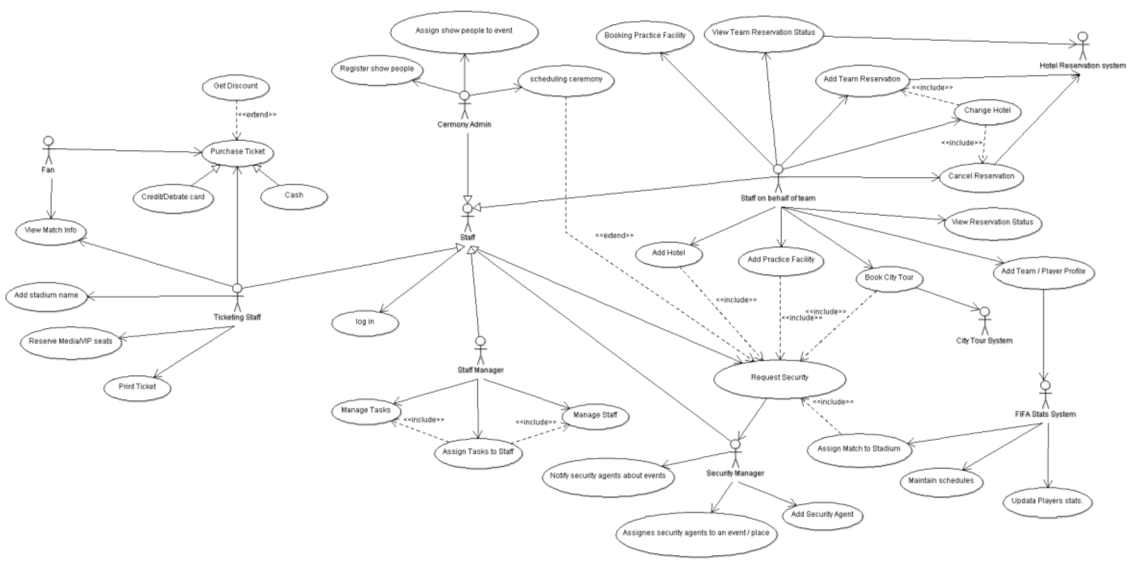


Figure 6 UML Use Case diagram

2.1.5. Semantic Transparency

The principle of Semantic Transparency suggests the use of visual representations whose appearance is highly suggestive of their meaning. Using semantically transparent symbols reduces the cognitive load because they have built-in mnemonics: their meaning can be either perceived directly or easily learnt [8].

2.1.6. Dual Coding

According to Dual Coding theory [9], using a combination of text and graphics together makes information presentation more obvious than using either on their own. Graphical and textual information are encoded in separate systems in working memory. Consequently, referential connections between the two systems in working memory are strengthened. It needs to be emphasized that text should not replace graphics and it certainly should not be used as the only means to differentiate between graphical symbols (textual differentiation). Text should be used to complement graphics as a form of redundant encoding [8], [9].

2.1.7. Cognitive Integration

The principle of Cognitive Integration refers to the ease with which a notation allows its reader to integrate different information from different diagrams. This is especially important when different diagrams are used to represent a system, which is often the case. There are two aspects of cognitive integration: conceptual and perceptual integration. Conceptual integration allows a reader to assemble a coherent mental representation of the system as a whole from separate diagrams. Perceptual integration is concerned with features in a notation that allow its reader to effectively navigate from one diagram to the next. Navigation in this sense requires a notation to have features that provide information about orientation, route choice, route monitoring and destination recognition.

2.1.8. Graphic Economy

The Graphic Economy principle refers the number of graphical symbols in a notation. A large number of symbols in a notation reduce its cognitive effectiveness. The literature has identified an upper limit of around six categories for the human ability to discriminate between perceptually distinct categories [10]. It is advised to leverage the extra room identified from the assessment based on the Graphic Economy principle to resolve other issues in the notation, such as symbol overload. However, caution needs to be exercised when introducing new perceptually distinct symbols by utilizing the shape variable. As mentioned before, shape is the most influential visual variable. Studies have shown that humans recognize different shapes as constructs that have categorically different meanings.

2.1.9. Cognitive Fit

The principle of Cognitive Fit suggests the use of different visual dialects for different tasks and audiences. Some dialects can be made complex and suitable for advanced users while other dialects can be simplified and made suitable for novices. It’s important to maintain readability and comprehensibility especially in any requirements engineering artifact since they used in collaboration with non-technical stakeholders, who can provide valid critical feedback.

2.2. Feature Diagrams

Software product line engineering is concerned with all development aspects for producing a set of related products that share more commonalities than variations [11]. Software product lines are emerging as an effective development paradigm that enables flexible response and mass customization [12]. Mass customization is about “producing goods and services to meet individual customer’s needs with near mass production efficiency”, according to [13]. Mass customization is a critical factor for development success as traditional mass production lines no longer suffice market needs [12]. Software product line engineering enables mass customization in mass production development environments [11].

Feature models are commonly used to document features in a software product line at different levels of abstractions. A feature model consists of one or more feature diagrams, composition rules, issues and decisions, and a system feature catalogue [14]. Features in a feature diagram are represented hierarchically with different relationships amongst features. Feature diagrams provide a visual summary of the features in a software product line. Due to the absence of a formal standard for feature diagrams, a review of the literature was required to identify the most state-of-the-art and canonical notational constructs. Feature diagrams made its first appearance in the literature as part of the Feature Oriented Domain Analysis (FODA) method [14]. The original form of feature diagrams shown in Figure 7.

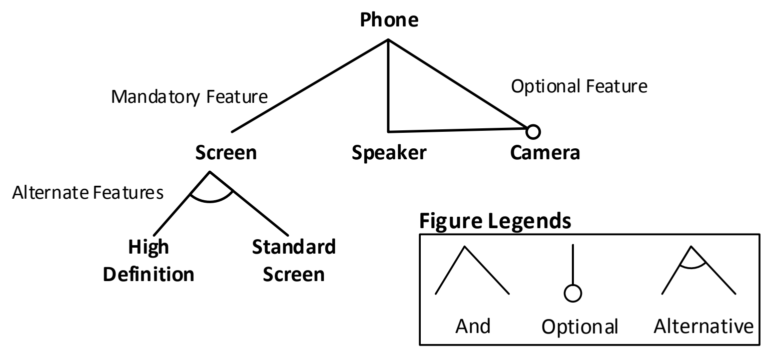


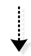
Figure 7 Original FODA Feature diagram

The notational set for feature diagrams has since been extended in many research works [11], [15]–[22], with the intention to increase the semantics covered by feature diagrams. Table 1 presents the notational set of feature diagrams and a brief definition of each symbol's semantics. Table 1 also shows the literature references where each symbol was first introduced. An example of feature diagram shows the entire notational set considered in this study is shown in Figure 8.

To the best of the author knowledge, the notational set described below is the most state-of-the-art and canonical notational constructs for feature diagrams.

Table 1 Feature diagrams semantics and symbols legend.

Semantic	Symbol	Explanation	Ref.
Concept		Symbolizes the software product line.	[17]
Feature		Decedent children nodes of the root.	[17]
Mandatory		Feature should be implemented whenever its parent is selected in the product	[15]
Optional		Feature could be optionally implemented whenever its parent is selected in the product.	[14]
And		Children feature could be selected when its parent feature is selected in the product.	[14]
alternative		Exactly only one feature of the children could be selected when its parent feature is selected in the product.	[14]
Or		One or more feature(s) of the children could be selected when its parent feature is selected in the product.	[15]

Semantic	Symbol	Explanation	Ref.
Require		If a feature A requires a feature B, the inclusion of A implies the inclusion of B.	[16]
Exclude		If a feature A excludes a feature B, both features cannot be part of the same product.	[16]
Feature cardinality	**[n..m]**	Defines number of instances of features that could be part of the concept.	[20]
Group cardinality	**<n..m>**	Limiting the number of child features which they could be part of the concept when their father is selected.	[20]
Attribute		It’s an extension to a features to accommodate extra information	[18]
Dead Feature		Feature cannot be part of the concept due to modeling anomaly.	[11]
Refer feature		Special symbol for decomposing feature to sub-tree of features.	[19]
Generalization		Relationship between child and parent feature which is abstracting the commonalities and reduce the details among the child features.	[17]
Implementation		Relationship exists between two features when one of them must implement, or realize, the behavior specified by the other.	[17]

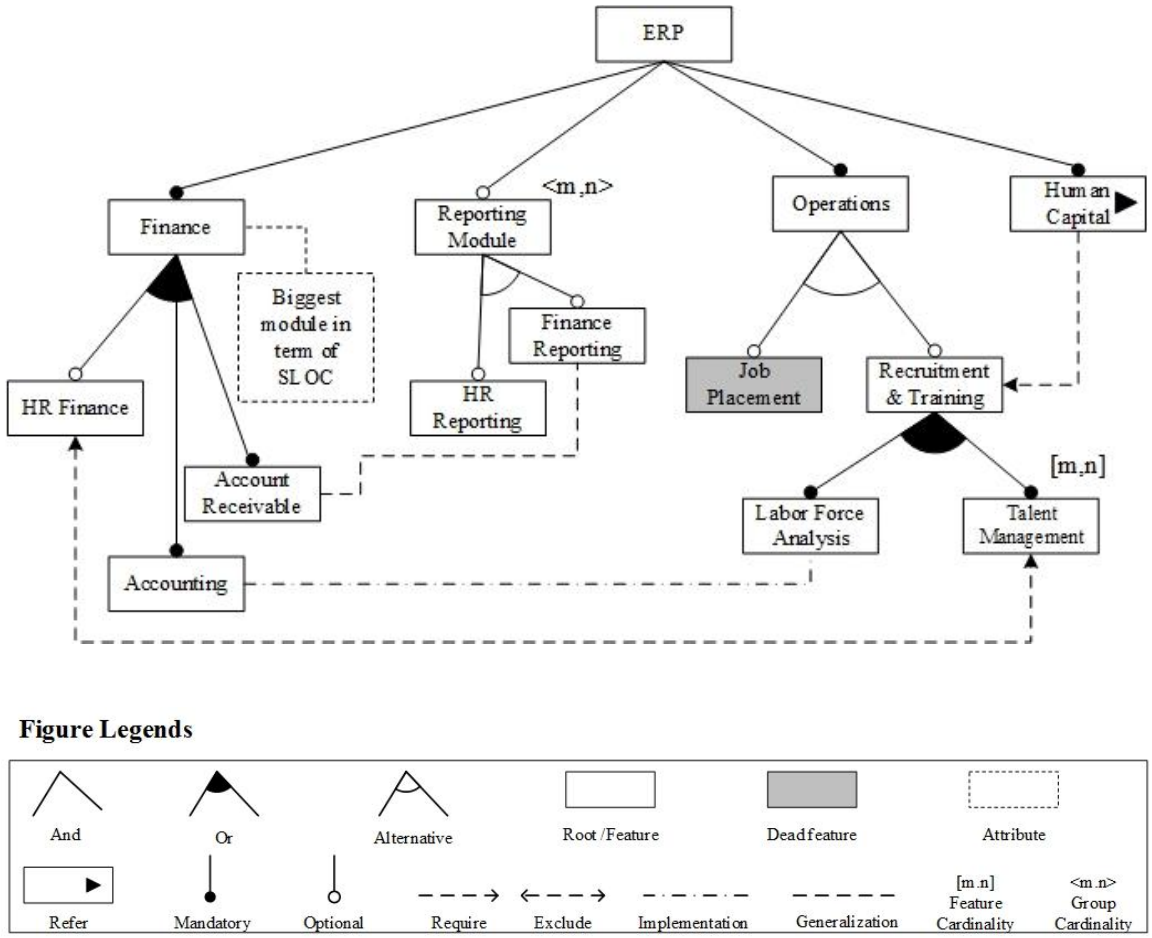


Figure 8 Example of a feature diagrams

2.3. Secure Tropos

Although the rapid advancement in the current software engineering practices, careful analysis shows that security requirements are often reflected at late stage of development [23]. Security considered as a key quality attributes that needs attention while building software system, security aspects dealing with authentication, access control, confidentiality, integrity and other crucial characteristics like capturing threats and consequently treat the vulnerability. One of the reasons resulting in delaying the incorporation of such important issue, is the traditional requirement engineering modeling techniques, they lack sufficient support specifying security requirements.

Secure Tropos [23] is an extension to the Tropos methodology [24] which is in turn based on i* modeling framework, ST can accommodate and account for security aspects during the whole development stages. The objective of Secure Tropos is to fulfill security goals of the system at any level of abstraction, therefore ST as a methodology allows wide range of stakeholders to participate in the modeling activity of the system-to-be. This flexibility of ST allows non-technical stakeholders to involve in the identification and acknowledgment of security concerns.

Originally, Tropos is a methodology for building agent software systems intended to serve different development phases; early requirement, late requirement, architectural design, detailed design and implementation phases. Developing of Secure Tropos aimed to provide a methodology that is not only tolerating developers to incorporate security requirements, but also to illuminate rations about them. ST contributions to the original Tropos methodology are in the enhancements made over i* framework, made it accommodate security concepts such as, security constraint, security dependency and security entity. These enhancements incorporate concepts of actors, goals, tasks, resources and social dependencies to define the responsibilities of actors. These communications produces number of diagrams, which serve as input to the succeeding development stage. Modeling activities of ST includes secure reference modeling, security constraint modeling, secure capabilities modeling [23]. Following, Tropos / Secure Tropos semantics described briefly, visual symbols illustrated in Figure 9. Semantics and symbols were derived from [23], [24] and from a metamodel for Tropos defined in [25].

2.3.1. Semantics of Tropos/Secure Tropos:

Common (Tropos) actors & dependency semantics

- **Actor:** representing physical, social or software actor. Actor can also represent a role or a position or an agent.
- **Role:** representing abstract characterization for the behavior of a social actor within specific context or domain.
- **Position:** representing set of rules played by an actor.
- **Agent:** actors, which can be physical agents, such as a person, or software agents.
- **Hard Goal:** representing actors' interests.
- **Soft Goal:** representing actors' interest that has no cutting edge.
- **Task (Plan):** representing a way for accomplishing particular goal.
- **Resource:** representing information or physical entity.

- **Dependency:** a relationship between two actors which one of them is depending on the other for acquire a goal or deliver a resource or execute a plan.
- **Capability:** representing actor's ability of defining, choosing and executing a plan to achieve a goal.
- **Belief:** representing actor knowledge of the world.
- **Contribution Links:** Edges, which describes the relationship between goals in term of their contribution to each other (positively or negatively).
- **Decomposition:** Edges, which decomposes goals to sub-goals using AND/OR
- **Means-End:** Edges, identifies resources, goals and plans that provide a mean for satisfying a goal.

Security Constraints

- **Security Constraint:** representing a set of restriction related to security issue.

Secure Dependency

- **Secure Dependency:** introduce security constraints to the dependency, both depender and dependee must agree to the constraint in order for the dependency to be satisfied.

Secure Entity

- **Secure Goal:** representing actors' interests with respect of security.
- **Secure Task:** representing a way for accomplishing particular security goal.
- **Secure Resource:** representing information or physical entity that related to security.

- **Secure capability:** represents the ability of an actor to achieve a secure goal, execute a task and deliver a secure resource.

Security Reference

- **Security Feature:** representing security attributes that the system under development must incorporate. Such as integrity privacy etc.
- **Protection Objective:** representing set of principles or rules that help the attainment of security feature.
- **Security Mechanism:** representing the standard security methods that contribute to satisfying protection objectives.
- **Threat:** representing the conditions that cause problems that probably endanger security feature in a system.
- **Positive / Negative contribution link:** link edge between security reference concepts, where positive contribution used to indicate help to fulfill one node to another, while negative contribution indicate refusal of that.

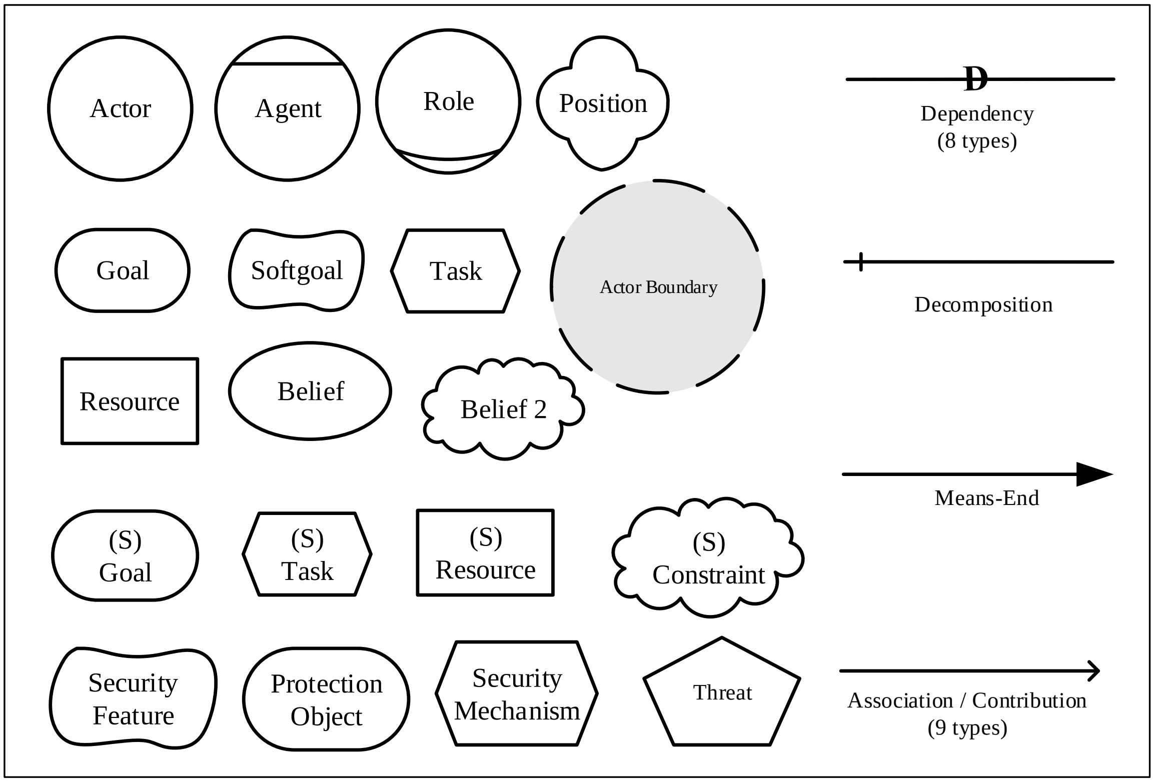


Figure 9 Visual symbols of Secure Tropos.

2.4. Related Work

Shortly after the publication of Moody's seminal paper [1], the area of visual notation evaluation has been increasingly gaining attention in the research community. In this exertion; an initial evaluation of visual syntax of the goal -oriented modeling language i* has been presented in [6]. Based on the principles defined in [1], the authors of [5] highlighted a number of shortcomings in the visual syntax of i*. In their ensuing work, the authors suggested various improvements to improve the semantic transparency of the i* visual syntax. John Thomas et al. also used the principles to design and improve cognitively effective business decision models [26]. A general evaluation of the UML (Unified Modeling Language) [27] suite of diagrams was presented in [4]. The study reveals that the design of the visual syntax of UML diagrams is not cognitively effective due to a lack of attention to visual aspects. Class diagrams in particular have been singled out as having the worst visual representation amongst UML diagrams [4]. The authors suggested general improvements that are applicable to all diagrams in UML. An evaluation of the visual syntax of the BPMN (Business Process Modeling Notation) [28] modeling language is presented in [29]. The notation of BPMN is expected to be understood by all stakeholders, however, the results of the evaluation revealed several shortcomings that hamper its comprehension by a subset of its stakeholders. The authors provided suggestions to improve the cognitive effectiveness of BPMN. In [30], the visual notation of use case maps [31] was evaluated. The evaluation shed light on several common weaknesses. The authors provided suggestions for improvements [30]. To the best of the author knowledge, no evaluation of the visual syntax of feature diagrams or secure Tropos based on Moody's principles was presented.

CHAPTER 3

ANALYSIS OF THE COGNITIVE EFFECTIVENESS

3.1. Analysis of Feature Diagram

3.1.1. Semiotic Clarity

As shown in Figure 1, the principle of Semiotic Clarity is at the center of the nine principles for cognitively effective notations, which is an indication of its primacy. The principle states that a visual language should have a one-to-one mapping between its symbols and the constructs they represent. One or more of the following anomalies are possible if a notation does not have the desired a one-to-one mapping:

Symbol Redundancy (synography) – multiple symbols can be used to represent one construct.

Symbol Overload (homography) – multiple constructs are represented by one symbol.

Symbol deficit – no symbol exists to represent a particular construct.

Symbol excess – a symbol that does not represent any construct.

- **Symbol Redundancy (synography)**

Symbol redundancy can lead to confusion as although choices are available to modelers, there is no basis for judgment. Symbol redundancy increases the burden on diagram readers as it requires them to memorize all the different symbols that can be used to represent one construct. Consequently, symbol redundancy increases the learning curve required to effectively use feature diagrams. However, feature diagrams do not contain such anomaly.

- **Symbol Overload (homography)**

Only one case of symbol overload exists in feature diagrams. The box symbol is used to represent 2 different constructs: root and feature. However, the same shape is used in dead feature, attribute and referring feature with slight differences which can negatively affect Perceptual Discriminability (see section 3.1.3). Symbol overload is perhaps the most dangerous of the four anomalies as it directly leads to confusion since there is no means to visually determine the semantic that is conveyed by a particular symbol [1].

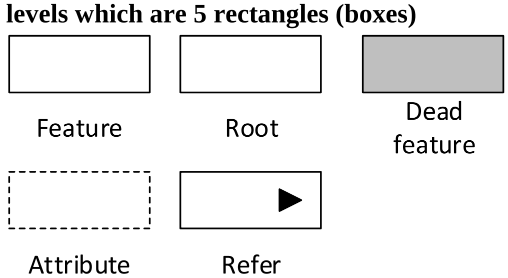
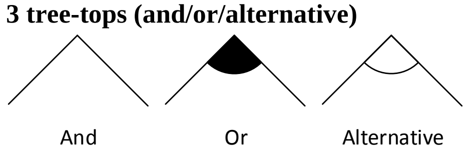
- **Symbol Deficit and Symbol Excess**

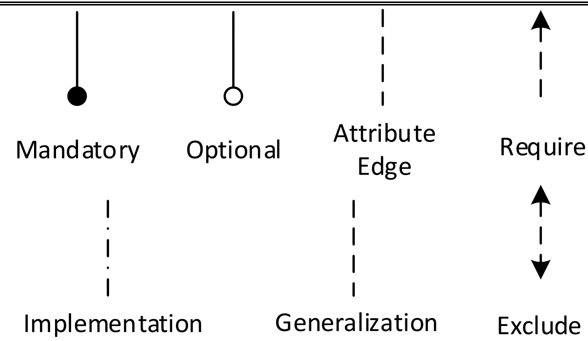
No symbol excess has been identified in the notational set of feature diagrams as each symbol represents at most one semantic construct. The only case of symbol deficit is concerned with the specification of feature and group cardinalities. Cardinality is only conveyed via textual annotation. Which is discouraged by principle of Dual Coding (see section 3.1.6).

3.1.2. Visual Expressiveness

The principle of Visual Expressiveness for a notation is concerned with the number of visual variables used and the ranges of values used in each variable. The larger the number of visual variables a notation uses and the wider the ranges of values used in each variable, the more visually expressive the notation becomes. Feature diagrams only use two visual variables: shape and brightness (see Table 2). With respect to the shape variable, feature diagrams use three values; rectangles, tree-tops and lines. With respect to brightness, feature diagrams use three levels: white, black and grey for shape fills. For edges, there is solid and two types of dotted lines. Further details of feature diagrams' use of the graphic space is presented in Table 2. Overall, the visual syntax of feature diagrams makes very limited use of the graphic space and hence cannot be considered visually expressive.

Table 2 Visual expressiveness of the feature diagrams notation.

<table>
<tr><th>Alphabet</th><th>Usage</th></tr>
<tr><td rowspan="3">Shape</td><td>levels which are 5 rectangles (boxes)

</td></tr>
<tr><td>3 tree-tops (and/or/alternative)

</td></tr>
<tr><td>7 lines (mandatory, optional, generalization, implementation, require, exclude and attribute line).</td></tr>
</table>



Brightness	**Low, high and medium, also dashed edges with two levels of dashes.**

3.1.3. Perceptual Discriminability

The principle of Perceptual Discriminability refers to the ease by which symbols from one notational set can be differentiated from each other. This principle requires the language designers to increase the **visual distance** [1] between symbols within a notational set. Visual distance is increased by utilizing a large number of visual variables and utilizing a wider range of values for each variable. Larger visual distance

between symbols allows them to be easily differentiated from each other. In this analysis we highlight symbols that are similar and thus have low levels of perceptual discriminability. Here similarity is determined by the *shape* variable since it has the greatest influence on cognition by humans [1]. Based on the findings from Section 3.2, three shape categories are identified: boxes, tree-tops and lines. A discussion of the perceptual discriminability of symbols within each shape category is presented below:

Boxes

This shape includes the *Root*, *feature*, *dead feature*, *refer feature* and *attribute* constructs. The brightness visual variable is the predominant variable used to distinguish symbols that use box shapes. The *root* and *feature* constructs are visually identical. *Dead features* are distinguishable by a grey color filling, while the *attribute* can be differentiated by a dotted border. The *refer feature* is distinguishable by a small "play-button" symbol inside the rectangle.

Tree-tops

This shape includes the *and*, *or*, and *alternative* decompositions. The *or* and *alternative decomposition* symbols can be differentiated by their brightness. The *or* and *alternative decomposition* symbols have black and white fillings, respectively. The *and decomposition* symbol can be differentiated by the absence of an upwards arc.

Lines

This shape includes the *mandatory*, *optional*, *generalization*, *implementation*, *require*, *exclude* and *attribute* line edges. The *generalization* and *attribute* symbols are identical while the *implementation* symbol is differentiated by coarse line brightness. The *require* and *exclude* symbols are very similar. The *require* symbol has one arrowhead at one end, while the *exclude* symbol has two arrow heads at both its ends. The *mandatory* and *optional* edges can be distinguished from each by their brightness. The *mandatory* and optional edges have black and white fillings, respectively.

It can be concluded that the visual syntax of feature diagrams overall suffers from low levels of perceptual discriminability.

3.1.4. Complexity Management

The principle of Complexity Management is described as *"the ability of a visual notation to represent information without overloading the human mind"* [1]. The presence of mechanisms that support the handling of complexity is essential for visual notations. High levels of complexity can severely limit model comprehension especially amongst novices. Complexity can be dealt with through modularization and hierarchical structuring. Modularity is a common practice to break down complexity by dividing large components into smaller ones. The notation of feature diagrams offers no mechanisms for complexity management via modularization. Hierarchal structuring manages complexity by providing representations of different levels of details and abstraction. Feature diagrams manage complexity by decomposing features into sub-features using the refers notation (hierarchal structuring).

3.1.5. Semantic Transparency

The principle of Semantic Transparency suggests the use of visual representations whose appearance is highly suggestive of their meaning. Using semantically transparent symbols reduces the cognitive load because they have built-in mnemonics: their meaning can be either perceived directly or easily learnt [30]. The visual syntax of feature diagrams cannot be considered as semantically transparent. Users of feature

diagrams are required to memorize the semantics of the symbols prior to reading or creating feature diagrams as they cannot infer the meanings of symbols simply by viewing them.

3.1.6. Dual Coding

According to Dual Coding theory [9], using a combination of text and graphics together makes information presentation more obvious than using either on their own. Graphical and textual information are encoded in separate systems in working memory. Consequently, referential connections between the two systems in working memory are strengthened. It needs to be emphasized that text should not replace graphics and it certainly should not be used as the only means to differentiate between graphical symbols (textual differentiation). Text should be used to complement graphics as a form of redundant encoding [8], [32].

Feature diagrams make no use of text as a form of redundant encoding. However, text has been combined with symbols in [11] for the require, exclude and extend edges. While [21] showed the same dual coding for the require and extend edges. This use of dual coding however is not consistent amongst the feature diagram modeling literature and hence were not considered in this evaluation.

3.1.7. Cognitive Integration

The principle of Cognitive Integration refers to the ease of which a notation allows its reader to integrate different information from different diagrams. This is especially important when different diagrams are used to represent a system, which is often the case. There are two aspects of cognitive integration: conceptual and perceptual integration. Conceptual integration allows a reader to assemble a coherent mental representation of the system as a whole from separate diagrams. Perceptual integration is concerned with features in a notation that allow its reader to effectively navigate from one diagram to the next. Navigation in this sense requires a notation to have features that provide information about orientation, route choice, route monitoring and destination recognition.

Feature diagrams do not offer any such mechanisms that support either type of cognitive integration. In [33], the authors presented a mechanism that allows the integration of feature diagrams with other UML diagrams. However, the approach is based on a model transformation technique that maps metaclasses of a proposed metamodel for feature diagrams with metaclasses of UML diagrams. This means that the approach is based on the metaclasses (semantics) of feature diagrams rather than directly based on its visual notation.

3.1.8. Graphic Economy

The Graphic Economy principle refers the number of graphical symbols in a notation. A large number of symbols in a notation reduce its cognitive effectiveness. The literature has identified an upper limit of around six categories for the human ability to discriminate between perceptually distinct categories [10]. Although the number of symbols in feature diagrams is far greater than six, the number of perceptually distinct symbols was determined to be only 3: boxes, tree-tops and lines (see Section 3.3). It is advised to leverage the extra room identified from the assessment based on the Graphic Economy principle to resolve other issues in the notation, such as symbol overload (see Section 3.1). However, caution needs to be exercised when introducing new perceptually distinct symbols by utilizing the shape variable. As mentioned before, shape is the most influential visual variable. Studies have shown that humans recognize different shapes as constructs that have categorically different meanings [5]. For feature diagrams, the semantic constructs of a *root*, *feature*, *dead feature*, *attribute* and *referring feature*, are not categorically different. They are all types of features. Therefore, it will be ill advised to use different shapes to represent these different semantics. Not to be deterred by this limitation, perceptual discriminability can be increased via using additional visual variables, such as color, texture and size, and textual encoding as a form of redundant encoding.

3.1.9. Cognitive Fit

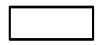
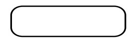
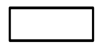
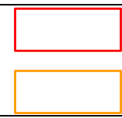
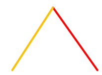
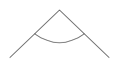
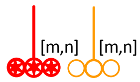
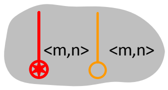
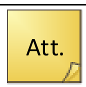
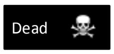
The principle of Cognitive Fit suggests the use different visual dialects for different tasks and audiences. Some dialects can be made complex and suitable for advanced users while other dialects can be simplified and made suitable for novices. However, feature diagrams have only one dialect, which is the case with most Software Engineering notations. Such drawback is referred to as "monolingualism" [1].

3.2. Suggested Improvement to Feature Diagram

The results of the notation evaluation indicate a number of suboptimal design aspects in the visual syntax of feature diagrams, thus reducing its cognitive effectiveness. In this section we suggest some enhancements based on deficiencies identified in Section 3.1. A summary of the suggested improvements presented in Table 3.

Symbol overload and perceptual discriminability are an outstanding issue in the notation of feature diagrams. Visual distance is increased in similar notations that categorized under boxes, tree-tops and edges. We suggest a new symbol for feature and group cardinality instead of only using textual annotation, these symbols will remove ambiguity and increase the cognitive effectiveness. Figure 10 illustrate proposed changes based on previous Figure 8 from feature diagram background (section 2.2).

Table 3 Suggested improvement for feature diagram.

Semantic	Symbol	Suggested	Justification
Root			Slightly changing the shape to increase the visual distance between the root and features
Feature			Change the color of the box according to its status if it's either mandatory or optional
Mandatory			Distance is increased by making mandatory colored red and optional with dark yellow. Mandatory is further distinguished by its strict symbols filled inside the circle
Optional			
And			Edges are colored to match the feature "mandatory – optional" status
Alternative			Arc is colored dark yellow to match the "optionality" behavior. Edges are colored to match feature status (not necessary dark yellow)
Or			Arc sector is filled with dark yellow to indicate its optionality behavior found in optional and alternative. Edges are colored to match feature status (not necessary dark yellow)
Require			Visual distance between require and exclude is increased by making require colored green and exclude blue, colors varies to avoid possible confusion with others edges of mandatory, optional, etc.
Exclude			
Feature cardinality	**[n,m]**		Feature cardinality symbol, colors and shape are made to match feature mandatory or optional status introduced earlier
Group cardinality	**<n,m>**		Group cardinality is differentiated by adding darker background on the effected features
Attribute of feature			Attribute is changed to note symbol
Dead Feature			Dead feature brightness now is low, additionally the skull symbol is introduced
Refer feature			Blue color of refer, underlined text and symbols are inspired by

Semantic	Symbol	Suggested	Justification
			hyperlinks vogue
Generalization			End arrow is changed by increasing the arrowhead to increase the visual distance between the generalization and implementation
Implementation			End arrow is changed, by decreasing the arrowhead to increase visual distance between generalization and implementations. Also the dash line is changed to increase visual distance

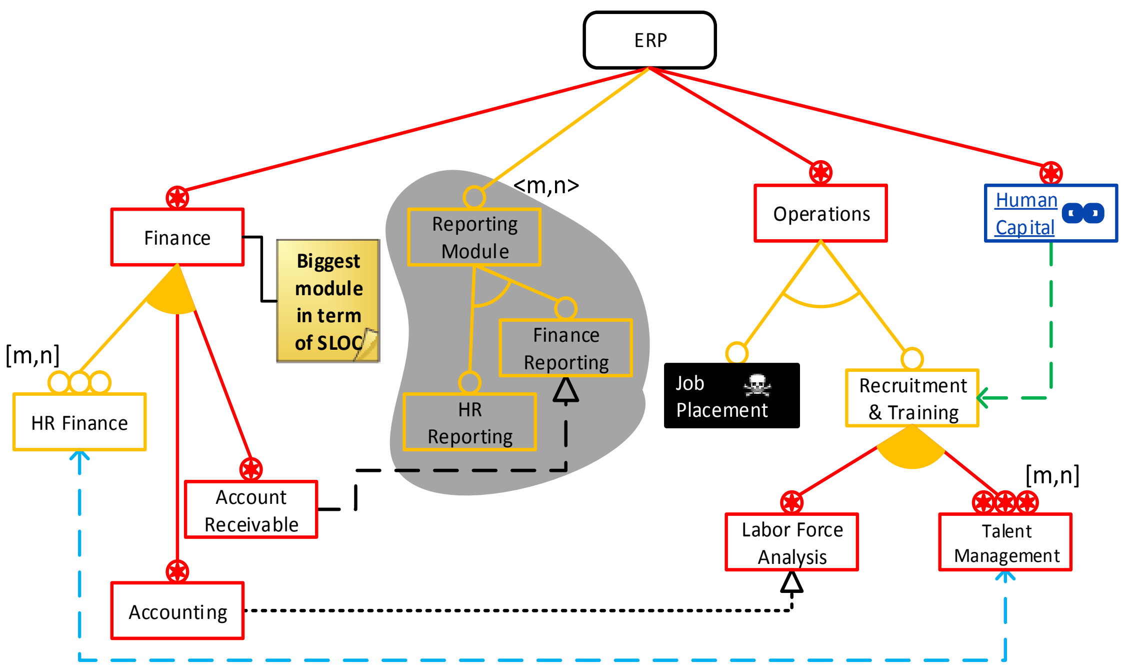


Figure 10 Feature diagram new notation.

3.3. Analysis of Secure Tropos

3.3.1. Semiotic Clarity

Secure Tropos has 40 different semantics, divided equally between 19 types of relations and 19 and types of elements (there is two semantics which has no symbols, capability and secure capability). These semantics represented by 16 distinct shapes, resulting in 17 symbol short. Table 4 summarizes semiotic clarity analysis.

Table 4 Summary of Secure Tropos semiotic clarity

Number of constructs	**40**
Number of distinct symbols	**16**
Symbol balance	**-24 (shortage)**
Number of symbol redundancy	**1**
Number of symbol overload	**29**
Number of symbol excess	**0**
Number of symbol deficit	**2**

Symbol Redundancy (synography)

There is only one incidence or symbol redundancy in the construct of believes (Figure 11). Believe could be modeled using two symbols, the cloud and oval shape

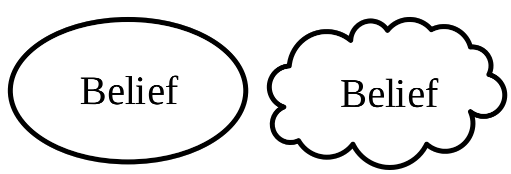


Figure 11 Symbol redundancy in Secure Tropos.

Symbol Overload (homography)

Secure Tropos has 29 different symbol overload, 16 are in relationships (8 dependency, 9 association/ contribute), and other 12 entity overloads. This result in 4 distinct shapes for relations and 5 distinct shapes for entities. This high number in overloads is due to usage of relations (association /contribution) and dependencies in different contexts. Moreover, Secure Tropos uses text as a differentiation mechanism of these symbols. Following Table 5 shows list of overloads in relationships and Figure 12 shows the symbols overload in entities.

Table 5 Overload in relationship symbols in Secure Tropos.

Symbols	Semantics of relations	Overload
	**Dependency:** **Resource dependency** **Plan dependency** **Goal dependency** **Actor dependency** **Secure Dependency:** **Secure Resource dependency** **Secure Plan dependency** **Secure Goal dependency** **Secure Actor dependency**	**8**
	**Decomposition**	**0**
	**Mean-End**	**0**
	**Relationships** **Association** **Has-A** **Want** **Execute** **Play** **Occupy** **Cover** **Own** **Contribution** **Positive contribution** **Negative contribution**	**9**

Symbols	Semantics of relations	Overload
Total = 4	Total = 19	Total = 17

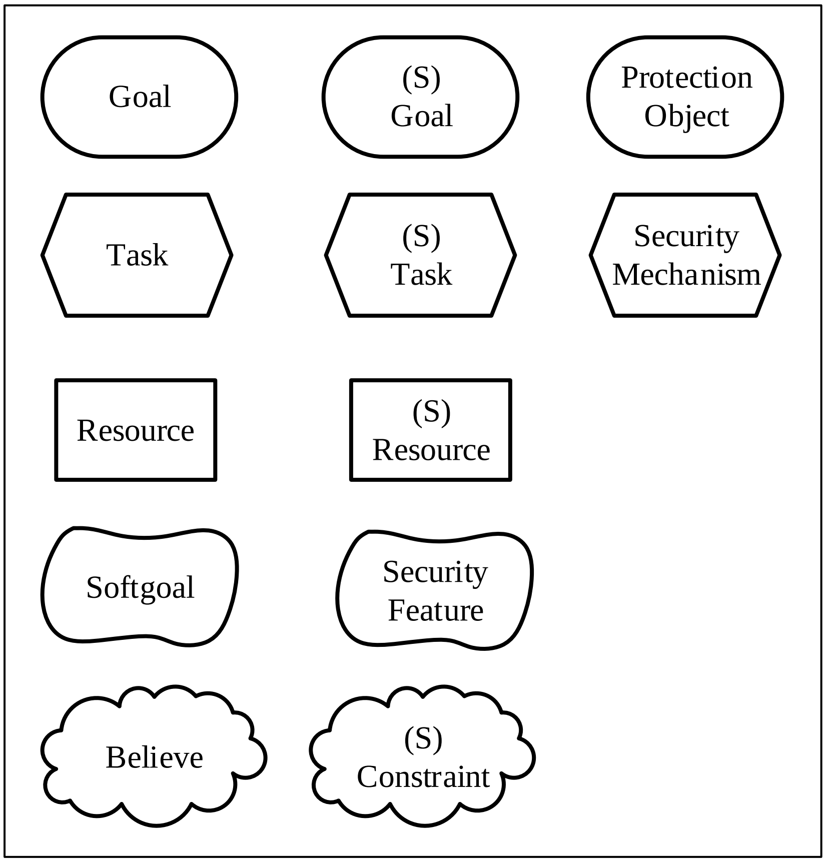


Figure 12 Symbols overload in Secure Tropos entities.

Symbol Excess

There is no symbol excess identified in the evaluation.

Symbol Deficit

Two constructs (Capability & Secure Capability) are not represented by any symbol. However, Secure Tropos presented these two construct in different diagrams (UML Activity diagram). Capability & Secure Capability presence in such diagrams will negate the existence of symbols deficit in Secure Tropos.

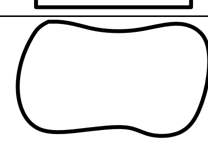
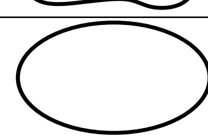
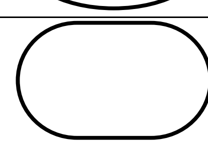
3.3.2. Visual Expressiveness

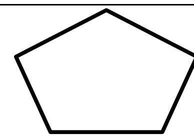
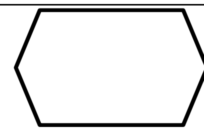
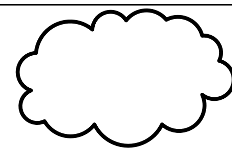
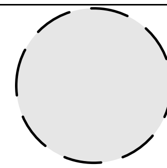
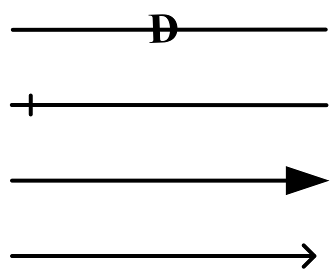
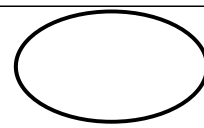
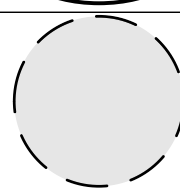
Visual Expressiveness (visual variables), uses 2 different variables (shape and brightness) in Secure Tropos. It is desirable characteristic to have as high as possible value of visual variables to increase visual expressiveness. With respect to the shape variable, Secure Tropos use 9 values; rectangles, curved rectangle, ovals, rectangular oval, pentagon, Hexagon, cloud, circle and lines. With respect to brightness, Secure Tropos uses two levels: high brightness and gray (medium) brightness in shapes, additionally there is solid and dotted lines. Further details of Secure Tropos usage of the graphical space presented in Table 7 while Table 6 summarizes the Visual Expressiveness of Secure Tropos. In conclusion, the visual syntax of Secure Tropos makes adequate use of the graphic space.

Table 6 Visual Expressiveness in Secure Tropos.

Alphabet	Usage
Shape	9 levels, 2 rectangles, 2 curved rectangles, 3 rectangular ovals, 1 oval, 1 pentagon, 3 hexagons, 2 cloud, 1 circle and 4 lines.
Brightness	High and medium, also dashed edges with two levels of dashes (solid & dotted).

Table 7 Secure Tropos usage of the graphical space.

Alphabet	Visual value	Symbol
Shape (9)	**Rectangles (2)**	
	Curved Rectangle (2)	
	Ovals (1)	
	Rectangular Oval (3)	

Alphabet	Visual value	Symbol
	Pentagon (1)	
	Hexagon (3)	
	Cloud (2)	
	Circle (1)	
	Lines (4)	
Brightness (2)	**High Brightness Solid Line**	
	Medium (gray) Brightness Dashed Line	

3.3.3. Perceptual Discriminability

This analysis highlights symbols which are similar and accordingly have low level of perceptual discriminability. Shape has the biggest impact over the similarity. Based on the findings from the previous section, three shape categories are identified: 8

geometrical shapes and 4 types of lines (see section 3.2.2 and Table 7). A discussion of the perceptual discriminability of symbols within each shape category presented below:

- **Geometrical shapes**

These shapes include:

  - Identical *Rectangles* for **Resource** and **Secure Resource**.
  - Identical *Curved Rectangles* for **Softgoal** and **Security Feature**.
  - Identical *Rectangular Ovals* used for **Goal**, **Secure Goal** and **Protection Object**.
  - Identical *Hexagons* used for **Task**, **Secure task** and **Security mechanism**.
  - Identical *Clouds* (Believe and Constraint).
  - An *Oval* used for **Believe**.
  - A *Pentagon* used for **Threat**.
  - A *Circle* used for **Actor Boundary**.

Usage of high brightness and solid shape-line is dominant in Secure Tropos notation, only one exception is in the case of actor boundary, which uses gray background and dashed container of circle shape.

- **Lines**

This shape includes dependencies, decompositions, mean-end and association/ contribution relationships. Dependencies (secure and normal dependencies) are notable because of the letter “D”. Decompositions are lines dividing goals to smaller sub goals, resulting in AND-Decomposition or OR-Decomposition. Both are straight lines with a small cross at one end. Mean-end is a line with filled arrow at one side. Association/ Contribution are text annotated lines are similar to mean-end but with smaller open arrow at the end.

As mentioned in semiotic clarity, Secure Tropos symbols (Figure 9 and Figure 13) differentiated using the context in which these symbols used, for example, resource dependency recognized when dependency incorporates a resource. Another differentiation mechanism uses textual annotation, which for example discriminate security entity over other entities using "(s)" as prefix to the entity name. Another example of textual annotation is with association / contribution lines, sometimes their semantics could not be inferred without textual annotation, since it is not trivial to novice users to spot their semantic using only contextual allocation, as an example, it is not easy for novices to tell when the association of cover take place (only between Position and Role). It can be concluded that the visual syntax of Secure Tropos overall suffers from low levels of perceptual discriminability.

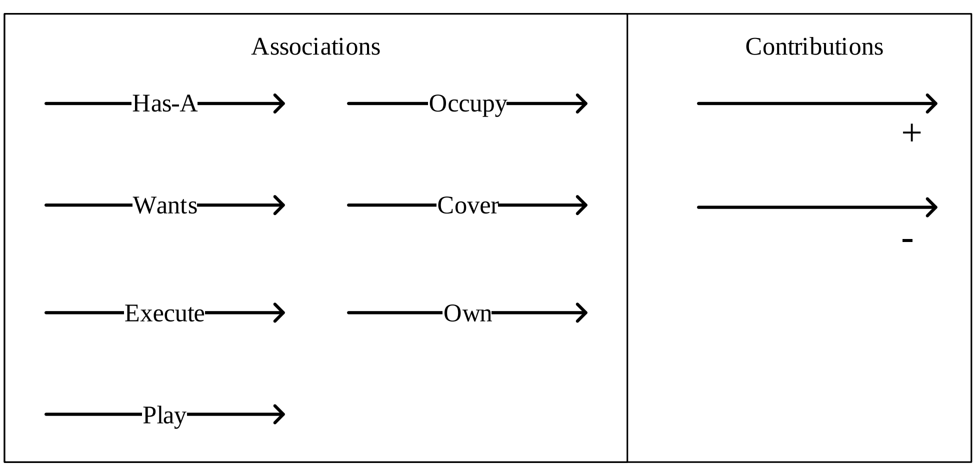


Figure 13 Textual Differentiation in Association/Contribution symbols.

3.3.4. Complexity Management

Secure Tropos did not incorporate a mechanism to break down complexity of the diagram, absent of explicit modularization and hierarchal decomposition will lead the reader of the diagram to process the diagram at once, such situation consequently will harshly limit model understanding among users**.**

3.3.5. Semantic Transparency

Secure Tropos lacks use of semantic transparency since geometric shapes and lines in abstract makes no inference of their meaning, Figure 14 shows actor symbol in Secure Tropos and in UML Use Case diagram. Users of Secure Tropos are required to learn and memorize visual construct along with their semantics to make use of the diagram.

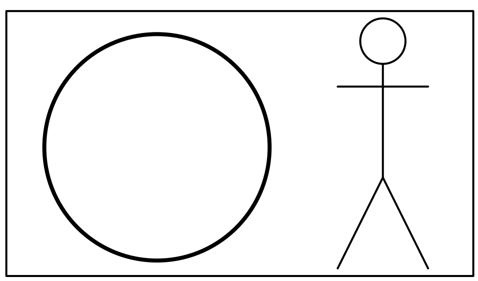


Figure 14 Actor representations in ST and UML. which makes sense?

3.3.6. Dual Coding

Dual coding suggest using of text and graphical symbols in a way that text is considered as a complement form of redundant coding, in other words text should not be the only differentiation among symbols. It is essential to strictly emphasize text should not substitute graphics and it certainly should not be used as the only means to differentiate between graphical symbols (textual differentiation). In this highlight, Secure Tropos make no use of dual coding, since symbols, especially Association / contribution line make heavy use of text, as it is the only differentiation mean of symbols semantic.

3.3.7. Cognitive Integration

Secure Tropos do not implement any mechanisms supports cognitive integration, although it has different modeling diagrams (actor diagram, goal diagram, capability diagram, security reference diagram, secure entities, and secure capability). Special note should be kept in consideration, capability/secure capability requires modeling using UML activity diagram and AUML diagram, Secure Tropos still lacks an explicit methodology to incorporate cognitive integration in this context.

3.3.8. Graphic Economy

The literature has identified an upper limit of around six categories for the human ability to discriminate between perceptually distinct categories [10]. Although the number of symbols in Secure Tropos is far greater than six, since the Secure Tropos visual vocabulary consisting of 8 distinct shapes and 4 types of lines, which they define graphic complexity of 12 for Secure Tropos notations.

3.3.9. Cognitive Fit

There is only one delegate for Secure Tropos. Principle of Cognitive Fit encourages using of different notation instances for different users and tasks.

CHAPTER 4

VALIDATION & EMPIRICAL STUDY

4.1. Subject-Based Empirical Evaluation

This section presents a subject-based experiment that was undertaken to study the performance of the proposed New Notation (NN) with respect to the Old Notation (ON) with respect of two dependent variables. The experiment performed in King Fahd University of Petroleum and Minerals, while the subjects were enrolled in Software Engineering undergraduate degree. The experiment is reported using the standard experimentation process presented by Wohlin et al. [34].

To determine the performance and the effects of the changes made to feature diagram notation, two dependent variables were observed. The dependent variables are shown in Table 8, 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 8 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)

Subjects involved in this experiment were Software Engineering undergraduate students enrolled in the second year of their degree. The experiment accomplished during the Spring term, (second semester of 2014 academic year). The participating subjects had already taken an introductory Software Engineering course that covered modeling in general. The subjects were taught lectures that covered Software Product Lines (SPL) and Feature models in detail. In a following classes, subjects performed exercises that further strengthen their knowledge of the Feature diagram modeling.

During the experiment, subjects were asked to infer feature diagrams of two systems, an Innovative Design system, and an ERP system. The diagrams are presented in Appendix A. The design of the diagrams satisfies two properties: (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 size in term of number of elements.

To perform the experiment, the subjects were divided randomly into two groups of 15 and 16 for Group A and Group B respectively. 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 [35]. Therefore, both versions of each diagram (NN and ON) were developed using an identical layout. A questionnaire (presented in Appendix A) was provided to the subjects that asked questions pertaining to the diagrams, such as “Label the following relationships using one of these terms: Require (R), Exclude (E), Implementation (I) and Generalization (G)”. The survey contains two identical sets of questions, one for each diagram. The participants asked to measure the time taken to answer questions related to one diagram at a time.

To analyze the results of the experiment, the famous statistical tests Mann-Whitney U statistic was used [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].

4.2. Results of the Empirical Evaluation

In this section the investigation of the effect of using the NN of feature diagram is compared against using the ON. This analysis provides the experimental results to provide some descriptive experiment-wide numerical analysis by considering the combined results from both Innovative design and ERP feature diagrams. The rationale of this analysis is to gain a wide-range evaluation of the overall impact of using the new notation and to gain further confidence to accept or reject the hypotheses of (T) and (E) variables.

Figure 15 shows the performance of groups A and B in aggregation ($n= 31$) with respect to the response times variable (T), while Figure 16 shows the performance of groups A and B in aggregation with respect to the errors committed variable (E). Descriptive statistics about the performances of groups A and B in aggregation are shown in Table 8. The results of the Mann-Whitney test (Table 9) indicate a statistical significance for both variables.

Table 8 Descriptive statistics for (Group A + Group B).

Variable	Notation	Min	1st Quartile	Median	3rd Quartile	Max	IQR
Response Times	NN	926	1658.5	1878.0	2175.5	3440	517.0
	ON	1215	2006.8	2236.0	2681.2	4807	674.3
Errors Committed	NN	0	1.0	3.0	4.8	66	3.8
	ON	0	4.0	6.0	11.8	147	7.8

Table 9 Mann-Whitney test for (Group A + Group B).

Variable	Notation	Rank sum	Mean rank	U	Difference between medians	95% CI	Mann-Whitney U statistic	P
Response Times	NN	740.5	23.89	716.5	-393.0	-∞ to -218.0	716.5	0.0004
	ON	1212.5	39.11	244.5				
Errors Committed	NN	732.5	23.34	733.5.0	-3.0	-∞ to -2.0	733.0	0.0002
	ON	1229.5	39.66	227.5				

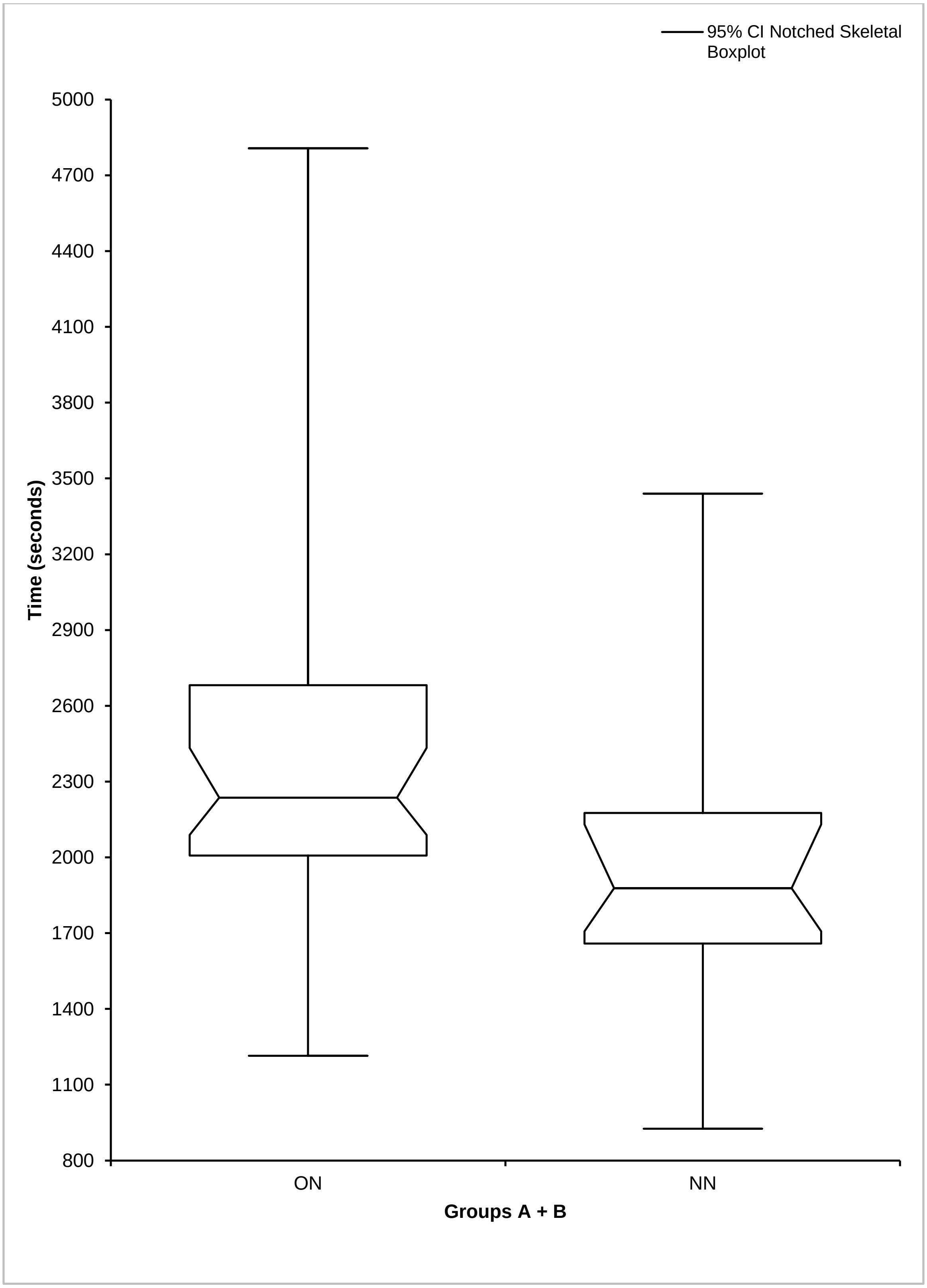


Figure 15 Group A + Group B performance with respect to response time and each feature diagram in isolation.

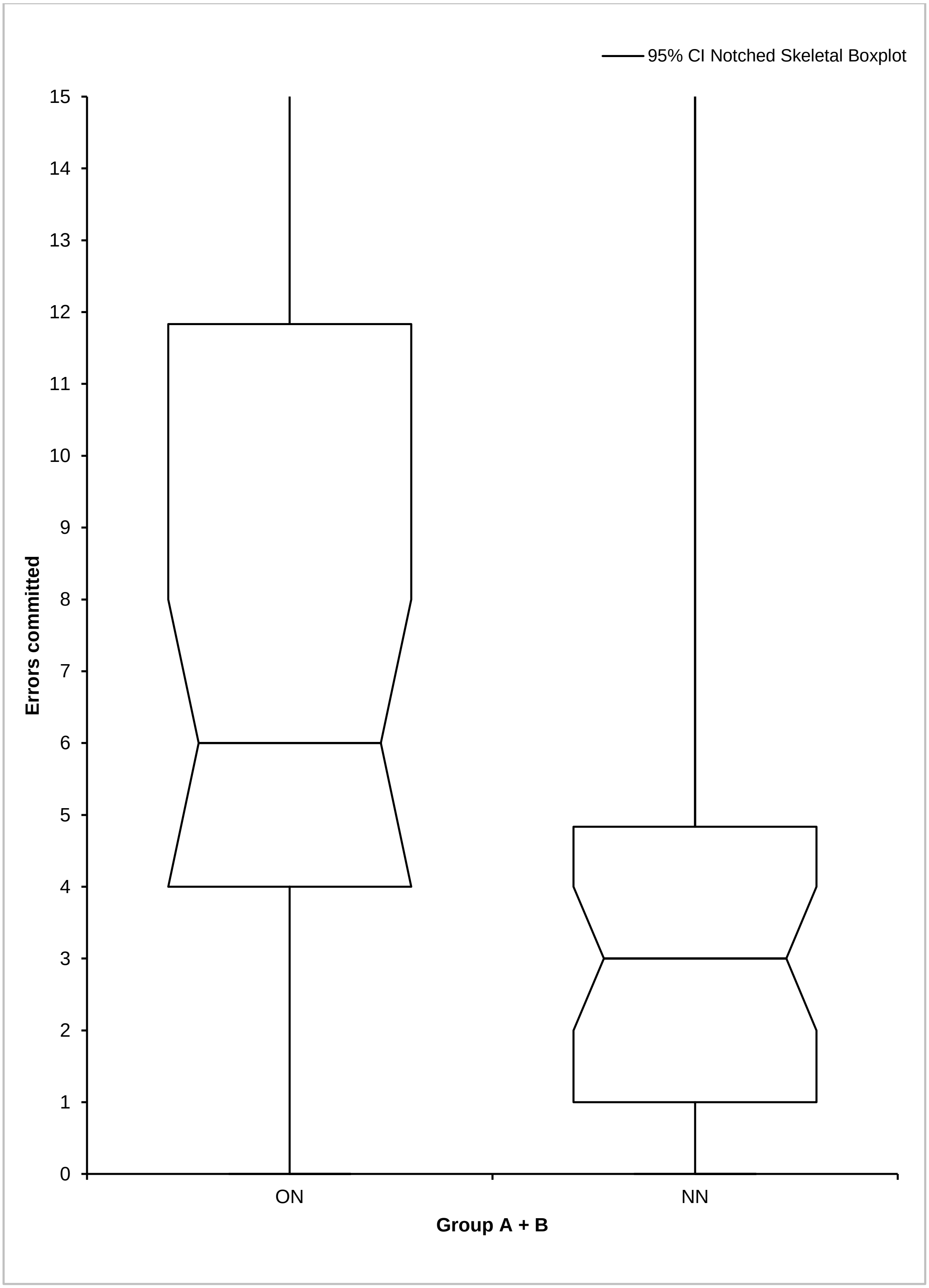


Figure 16 Group A + Group B performance with respect to error committed and each feature diagram in isolation.

4.3. Qualitative Analysis

At the end of both group A and group B students' questionnaire, they have been asked to answer three questions and leave their comments and feedback:

- First question was which notation did you like the most (Original or New)?
- Second question was why did you like the notation you chose? What was your experience when performing the exercises?
- Third question was how do you think the notation you chose can be improved?

Upon student completion of the survey, 25 student among of the 31 student chosen new notation as the their preferred one, this forms more than 80% of the participants, only 4 students (13%) preferred original notation - although one of them did actually liked some symbols of the new proposed notation. Participants whom chose new notation almost agreed that new notation is easier to read, take less time to identify and trace symbols, they also expresses there endorsements of the new notation symbols, specially relations (generalization, implementation, require and exclude) and group cardinality. Participants whom favored original notation mainly they liked it because of its' excessive use of colors.

Participants has suggested to further increase the discriminability in relations, namely they asked to increase distance between generalize and implement, they suggested used of colors, changing shape of arrows. Number of the participants showed their conservations about color selection, for example couple of students criticized usage of the unpleasant yellow in optional feature and they suggested usage of green instead.

CHAPTER 5

Conclusion

The cognitive effectiveness of Software Engineering notations has long been a neglected area of research. The notation of feature diagrams and secure tropos is no exception. This thesis presented a scientific evaluation of the visual syntax of feature diagrams and secure tropos. The evaluation revealed a number of suboptimal design aspects in the visual syntax of both diagrams. A set of improvements were suggested that can potentially overcome these design drawbacks and improve the overall cognitive effectiveness of feature diagrams. The improvements were conjured based on the same principles used to assess the visual syntax.

An empirical evaluation of the new suggested notation was performed on subjects to assess the effectiveness of the changes in comparison with old notation. Two depended variable were studied to assess the effectiveness of the new changes, Response Time (T) and Errors Committed (E). Response Time (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. Error Committed (T) for which the alternative hypothesis indicates that errors committed by the diagram readers in the NN will be less than the error committed by diagram reader in the ON. Results show statistical significance which allows the rejection of null hypotheses and accepting the alternative hypotheses, which means that the new notation outperformed the original notation in the two variables.

Future work can be directed towards providing automation support for the new notation of feature diagram. Automation support will greatly aid its adoption by potential users. Many necessary research works have been conducted that evaluates visual notations in Software Engineering. However, many more notations remain to be assessed. The improvements suggested in the literature with respect to various Software Engineering notations lack empirical evidence. Empirical validation of these improvements will increase the confidence to apply and standardize changes to existing notations.

Appendix

Evaluation of the suggested feature diagram notation

Two groups of students (group A n = 15, group B n = 16) were given a survey, asked to solve questions and give comments and feedback. Both groups are identical in terms of size and questions, Each group contains two diagrams, one with the old notation and the other with the new notation set, Group A diagram 1 was old notation while diagram two is the new notation. As for group B, diagram 1 was the new notation and diagram 2 is the old notation. This order guarantees that both groups of students (old and new diagrams) spent the same the learning curve.

--Start of the survey--

<u>RESET YOUR TIMER NOW</u>

<u>Feature Diagram 1</u>

Q1. Please count the number of:

____ Mandatory Features

____ Optional Features

____ Dead

____ Attribute

____ Refer

____ Feature Cardinality (mandatory)

____ Feature Cardinality (optional)

____ Group Cardinality

____ Require relations

____ Exclude relations

____ Implementation relations

____ Generalization relations

Q2. Please label the following features using one of these terms: Attribute (A), Dead (D), Refer (R), Mandatory (M), and Optional (O).

	Feature		Feature		Feature		Feature		Feature
_	3D Drawing	_	CSS Integration	_	Lasso Tool	_	Premium Audio Suite	_	Subtitle Creation
_	3D Printing	_	CSS & Macrovision	_	Local Lan	_	Printing Service	_	Test Script & Test Cases
_	86-bit Support	_	Delay & Echo	_	Lossless Format	_	RAID	_	Testing Environment
_	Adjust	_	Device Profile	_	Media Server	_	Regression Analysis	_	Text & Fonts
_	Apps Integration	_	Editing Sound	_	Media Server Integration	_	Regular printing	_	Text-image Recognition
_	Audio	_	Effects	_	Mixer	_	Remove Noise	_	This feature allow users to obtain new updates
_	Audio I/O	_	Exposure	_	Modulation	_	Render	_	This feature gives the ability to master audio files with variety of input devices
_	Binarization	_	File Sharing	_	Motion Picture	_	Reverb	_	Upgrade Mechanism
_	Blur	_	Filters	_	Noise Cancelation	_	Save Image	_	Vector Graphic
_	BluRay Burn	_	Hard Disk Driver	_	PDF Creator	_	Share Mobile Profile	_	Vector Image
_	Cloud	_	Histogram	_	Performance Testing	_	Skew Fix	_	WC3 Standard
_	Color Balance	_	Image Processing	_	PHP Programing	_	Smart Devices Emulator	_	Web Development
_	Compare Devices	_	Input Processor	_	PostScript	_	Smart Lasso Tool	_	XML Configuration
_	Contours Detects	_	JavaScript Integration	_	Pre Processing	_	Sound Processing		
_	Convert to Shape	_	Language Localization	_	Prebuild Setup	_	Standard Lasso Tool		

Q3. Label the following relationships using one of these terms: Require (R), Exclude (E), Implementation (I) and Generalization (G).

	Source Feature	Destination Feature
____	**Language Localization**	**XML Configuration**
____	**PostScript**	**PDF Creator**
____	**Vector Image**	**Vector Graphic**
____	**Printing Service**	**Cloud**
____	**Text-image Recognition**	**Binarization**
____	**Media Server Integration**	**Apps Integration**
____	**Sound Processing**	**Audio**
____	**CSS & Macrovision**	**86-bit Support**
____	**3D Printing**	**3D Drawing**
____	**Save Image**	**Media Server**
____	**Prebuild Setup**	**Hard Disk Driver**
____	**CSS & Macrovision**	**CSS Integration**
____	**Contours Detects**	**Text & Fonts**
____	**BluRay Burn**	**RAID**
____	**Share Mobile Profile**	**File Sharing**
____	**Testing Environment**	**Web Development**

Q4. Find features with cardinality. Label them either Mandatory Cardinality (M) or Optional Cardinality (O).

	Feature		Feature		Feature		Feature		Feature
_	3D Drawing	_	CSS Integration	_	Lasso Tool	_	Premium Audio Suite	_	Subtitle Creation
_	3D Printing	_	CSS & Macrovision	_	Local Lan	_	Printing Service	_	Test Script & Test Cases
_	86-bit Support	_	Delay & Echo	_	Lossless Format	_	RAID	_	Testing Environment
_	Adjust	_	Device Profile	_	Media Server	_	Regression Analysis	_	Text & Fonts
_	Apps Integration	_	Editing Sound	_	Media Server Integration	_	Regular printing	_	Text-image Recognition
_	Audio	_	Effects	_	Mixer	_	Remove Noise	_	This feature allow users to obtain new updates
_	Audio I/O	_	Exposure	_	Modulation	_	Render	_	This feature gives the ability to master audio files with variety of input devices
_	Binarization	_	File Sharing	_	Motion Picture	_	Reverb	_	Upgrade Mechanism
_	Blur	_	Filters	_	Noise Cancelation	_	Save Image	_	Vector Graphic
_	BluRay Burn	_	Hard Disk Driver	_	PDF Creator	_	Share Mobile Profile	_	Vector Image
_	Cloud	_	Histogram	_	Performance Testing	_	Skew Fix	_	WC3 Standard
_	Color Balance	_	Image Processing	_	PHP Programing	_	Smart Devices Emulator	_	Web Development
_	Compare Devices	_	Input Processor	_	PostScript	_	Smart Lasso Tool	_	XML Configuration
_	Contours Detects	_	JavaScript Integration	_	Pre Processing	_	Sound Processing		
_	Convert to Shape	_	Language Localization	_	Prebuild Setup	_	Standard Lasso Tool		

Q5. Find the group cardinality and label the group node with either Mandatory (M) or Optional (O).

	Feature		Feature		Feature		Feature		Feature
_	3D Drawing	_	CSS Integration	_	Lasso Tool	_	Premium Audio Suite	_	Subtitle Creation
_	3D Printing	_	CSS & Macrovision	_	Local Lan	_	Printing Service	_	Test Script & Test Cases
_	86-bit Support	_	Delay & Echo	_	Lossless Format	_	RAID	_	Testing Environment
_	Adjust	_	Device Profile	_	Media Server	_	Regression Analysis	_	Text & Fonts
_	Apps Integration	_	Editing Sound	_	Media Server Integration	_	Regular printing	_	Text-image Recognition
_	Audio	_	Effects	_	Mixer	_	Remove Noise	_	This feature allow users to obtain new updates
_	Audio I/O	_	Exposure	_	Modulation	_	Render	_	This feature gives the ability to master audio files with variety of input devices
_	Binarization	_	File Sharing	_	Motion Picture	_	Reverb	_	Upgrade Mechanism
_	Blur	_	Filters	_	Noise Cancelation	_	Save Image	_	Vector Graphic
_	BluRay Burn	_	Hard Disk Driver	_	PDF Creator	_	Share Mobile Profile	_	Vector Image
_	Cloud	_	Histogram	_	Performance Testing	_	Skew Fix	_	WC3 Standard
_	Color Balance	_	Image Processing	_	PHP Programing	_	Smart Devices Emulator	_	Web Development
_	Compare Devices	_	Input Processor	_	PostScript	_	Smart Lasso Tool	_	XML Configuration
_	Contours Detects	_	JavaScript Integration	_	Pre Processing	_	Sound Processing		
_	Convert to Shape	_	Language Localization	_	Prebuild Setup	_	Standard Lasso Tool		

Time Completed

Minutes	Seconds

RESET YOUR TIMER NOW
Feature Diagram 2

Q1. Please count the number of

____ Mandatory Features

____ Optional Features

____ Dead

____ Attribute

____ Refer

____ Feature Cardinality (mandatory)

____ Feature Cardinality (optional)

____ Group Cardinality

____ Require relations

____ Exclude relations

____ Implementation relations

____ Generalization relations

Q2. Please label the following features using one of these terms: Attribute (A), Dead (D), Refer (R), Mandatory (M), and Optional (O).

	Feature		Feature		Feature		Feature		Feature
_	Accounting	_	Distribution Logistics	_	HR Management	_	Marketing and Sales	_	Recycling
_	Accounts Receivable	_	Employee Benefits	_	HR Reporting	_	Operations	_	Regulatory Compliance
_	Attendance	_	Enterprise Risk Management	_	Human Capital	_	Order Fulfillment	_	Reporting Module
_	Auditing	_	Experimental Reporting	_	Income Tax	_	Pay raise	_	Returns Management
_	Barcodes Management	_	External	_	Industry Standard	_	Payroll	_	RFID Management
_	Biggest module in term of SLOC	_	External Auditing	_	Internal	_	Performance History	_	Risk Management
_	Business Process	_	Finance	_	Internal Auditing	_	Performance Rewarding	_	Subscribers Billing
_	Company Policy	_	Finance Reporting	_	Job Placement	_	Process Management	_	Supply Chain Management
_	Contractors Management	_	Financial Accounting	_	Labor Force Analysis	_	Product Catalogs	_	Talent Management
_	Corporate Governance	_	Financial Risk Management	_	Leaves	_	Product Development	_	This feature implemented in Java
_	Credit Card	_	GAAP Regulation	_	Liquidity Risk	_	Production Logistics	_	Track & Trace
_	Customer Contact Center	_	Government Auditing	_	Logistics	_	Promotions	_	Unsubscribe
_	Customer Relationship Management	_	Government Tax Regulation	_	Managerial Accounting	_	R & D	_	Wire Transfer
_	Daily Attendance	_	History Record	_	Manufacturin g Management	_	Raw Material		
_	Desk Support	_	HR Finance	_	Manufacturin g Process	_	Recruitment & Training		

Q3. Label the following relationships using one of these terms: Require (R), Exclude (E), Implementation (I) and Generalization (G).

	Source Feature	Destination Feature
____	**Risk Management**	**Contractors Management**
____	**Manufacturing Management**	**Logistics**
____	**Government Tax Regulation**	**Income Tax**
____	**HR Management**	**HR Finance**
____	**Payroll**	**Reporting Module**
____	**Human Capital**	**Recruitment & Training**
____	**Accounts Receivable**	**Finance Reporting**
____	**Subscribers Billing**	**Accounting**
____	**Auditing**	**Regulatory Compliance**
____	**Leaves**	**Company Policy**
____	**Desk Support**	**Performance History**
____	**Production Logistics**	**Industry Standard**
____	**Promotions**	**Labor Force Analysis**
____	**Financial Accounting**	**GAAP Regulation**
____	**Raw Material**	**History Record**
____	**Marketing and Sales**	**Daily Attendance**

Q4. Find features with cardinality. Label them either Mandatory Cardinality (M) or Optional Cardinality (O).

	Feature		Feature		Feature		Feature		Feature
_	Accounting	_	Distribution Logistics	_	HR Management	_	Marketing and Sales	_	Recycling
_	Accounts Receivable	_	Employee Benefits	_	HR Reporting	_	Operations	_	Regulatory Compliance
_	Attendance	_	Enterprise Risk Management	_	Human Capital	_	Order Fulfillment	_	Reporting Module
_	Auditing	_	Experimental Reporting	_	Income Tax	_	Pay raise	_	Returns Management
_	Barcodes Management	_	External	_	Industry Standard	_	Payroll	_	RFID Management
_	Biggest module in term of SLOC	_	External Auditing	_	Internal	_	Performance History	_	Risk Management
_	Business Process	_	Finance	_	Internal Auditing	_	Performance Rewarding	_	Subscribers Billing
_	Company Policy	_	Finance Reporting	_	Job Placement	_	Process Management	_	Supply Chain Management
_	Contractors Management	_	Financial Accounting	_	Labor Force Analysis	_	Product Catalogs	_	Talent Management
_	Corporate Governance	_	Financial Risk Management	_	Leaves	_	Product Development	_	This feature implemented in Java
_	Credit Card	_	GAAP Regulation	_	Liquidity Risk	_	Production Logistics	_	Track & Trace
_	Customer Contact Center	_	Government Auditing	_	Logistics	_	Promotions	_	Unsubscribe
_	Customer Relationship Management	_	Government Tax Regulation	_	Managerial Accounting	_	R & D	_	Wire Transfer
_	Daily Attendance	_	History Record	_	Manufacturing Management	_	Raw Material		
_	Desk Support	_	HR Finance	_	Manufacturing Process	_	Recruitment & Training		

Q5. Find the group cardinality and label the group node with either Mandatory (M) or Optional (O).

	Feature		Feature		Feature		Feature		Feature
_	Accounting	_	Distribution Logistics	_	HR Management	_	Marketing and Sales	_	Recycling
_	Accounts Receivable	_	Employee Benefits	_	HR Reporting	_	Operations	_	Regulatory Compliance
_	Attendance	_	Enterprise Risk Management	_	Human Capital	_	Order Fulfillment	_	Reporting Module
_	Auditing	_	Experimental Reporting	_	Income Tax	_	Pay raise	_	Returns Management
_	Barcodes Management	_	External	_	Industry Standard	_	Payroll	_	RFID Management
_	Biggest module in term of SLOC	_	External Auditing	_	Internal	_	Performance History	_	Risk Management
_	Business Process	_	Finance	_	Internal Auditing	_	Performance Rewarding	_	Subscribers Billing
_	Company Policy	_	Finance Reporting	_	Job Placement	_	Process Management	_	Supply Chain Management
_	Contractors Management	_	Financial Accounting	_	Labor Force Analysis	_	Product Catalogs	_	Talent Management
_	Corporate Governance	_	Financial Risk Management	_	Leaves	_	Product Development	_	This feature implemented in Java
_	Credit Card	_	GAAP Regulation	_	Liquidity Risk	_	Production Logistics	_	Track & Trace
_	Customer Contact Center	_	Government Auditing	_	Logistics	_	Promotions	_	Unsubscribe
_	Customer Relationship Management	_	Government Tax Regulation	_	Managerial Accounting	_	R & D	_	Wire Transfer
_	Daily Attendance	_	History Record	_	Manufacturing Management	_	Raw Material		
_	Desk Support	_	HR Finance	_	Manufacturing Process	_	Recruitment & Training		

Time Completed

Minutes	Seconds

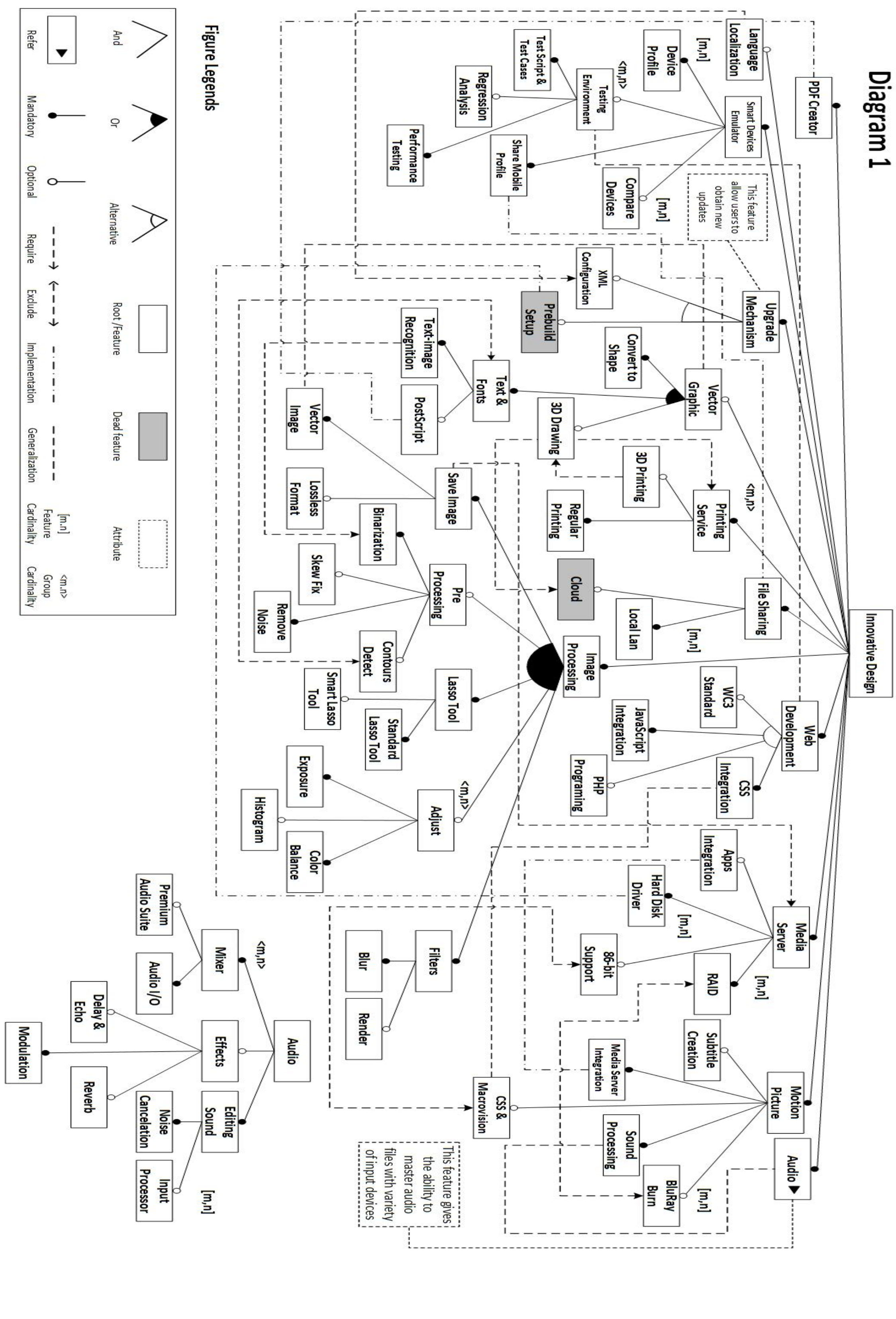
Questions:

Q1- Which notation did you like the most (original or new)?

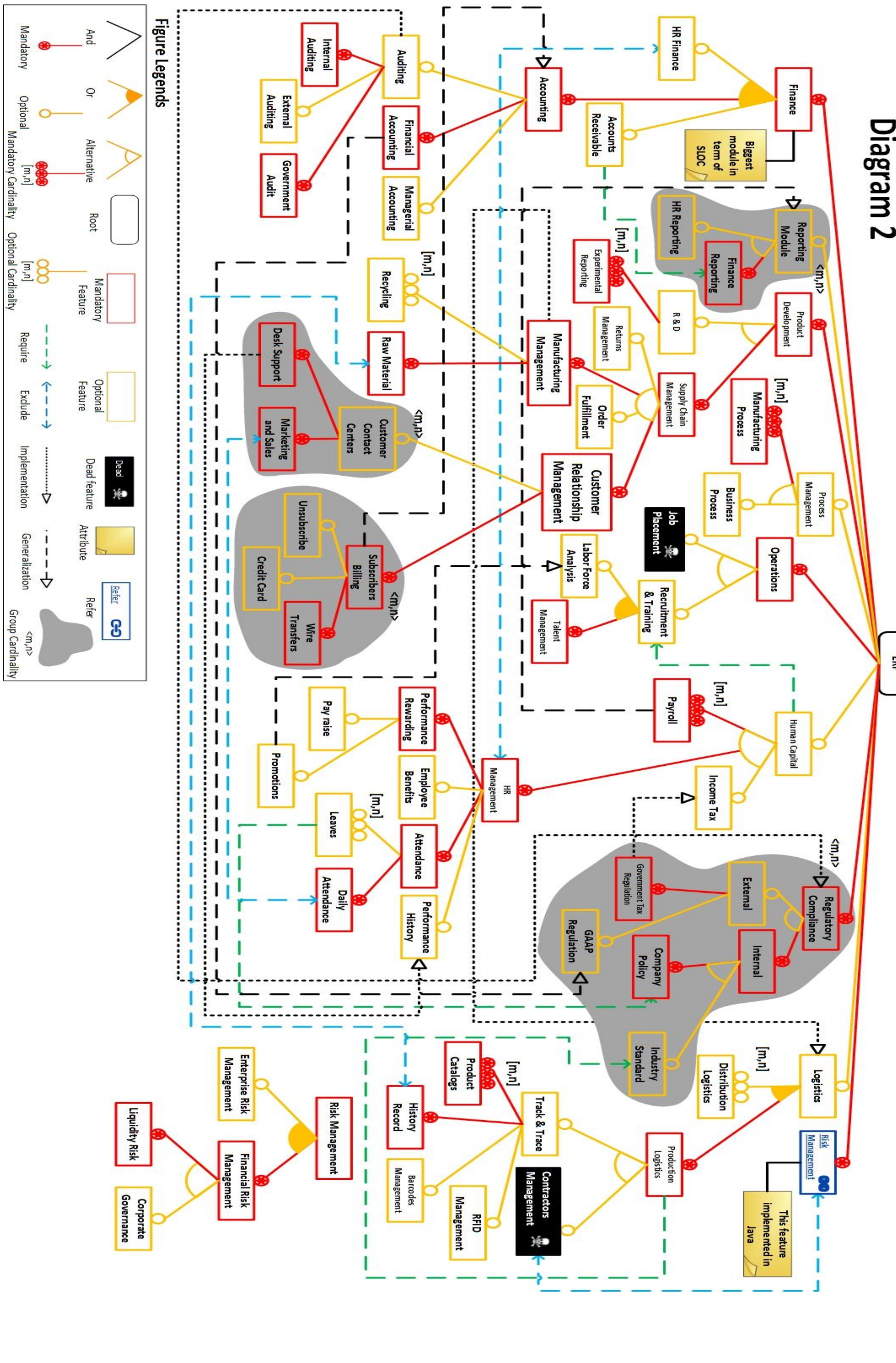
Q2- Why did you like the notation you chose? What was your experience when performing the exercises?

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

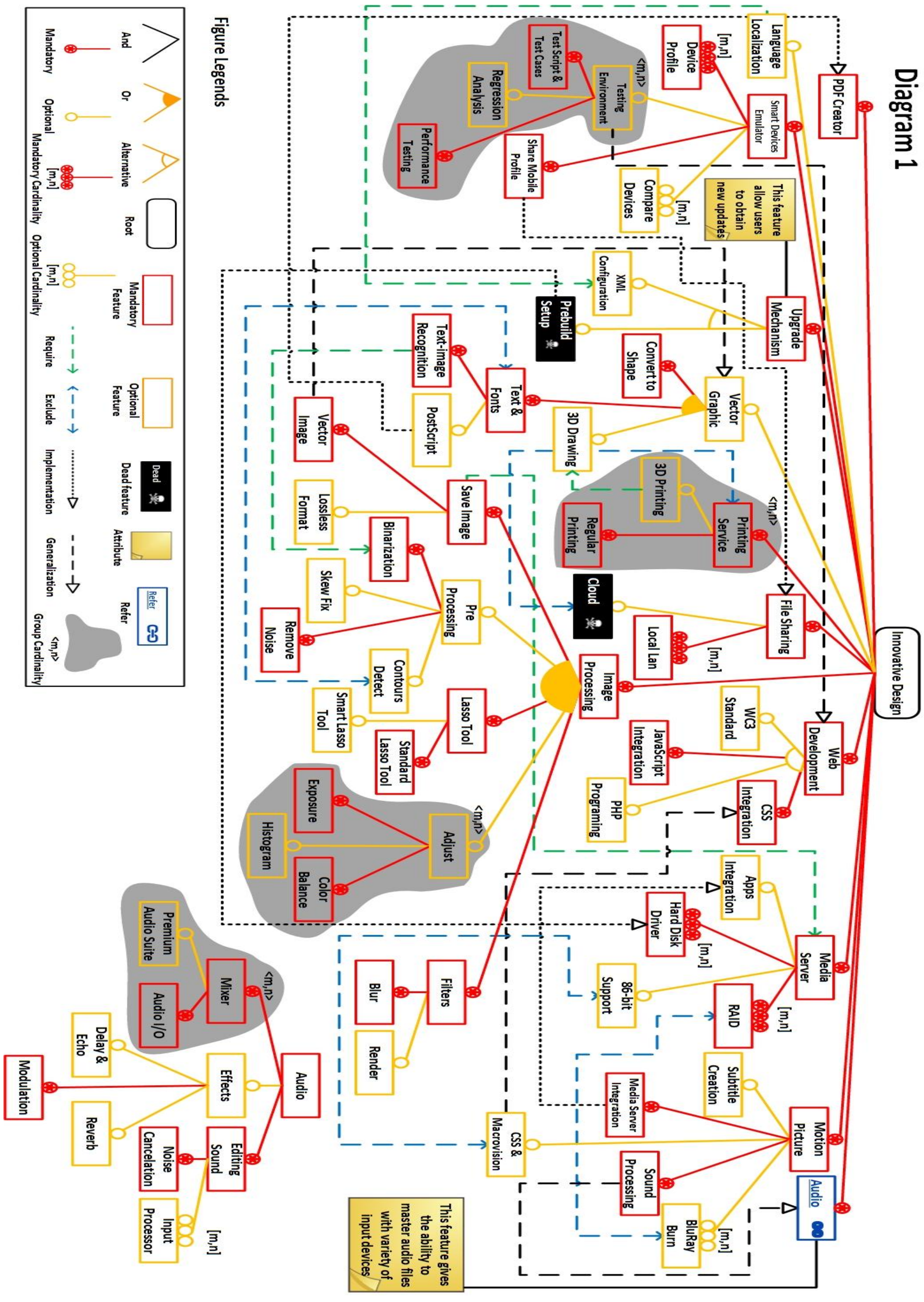
Group A, Diagram 1:



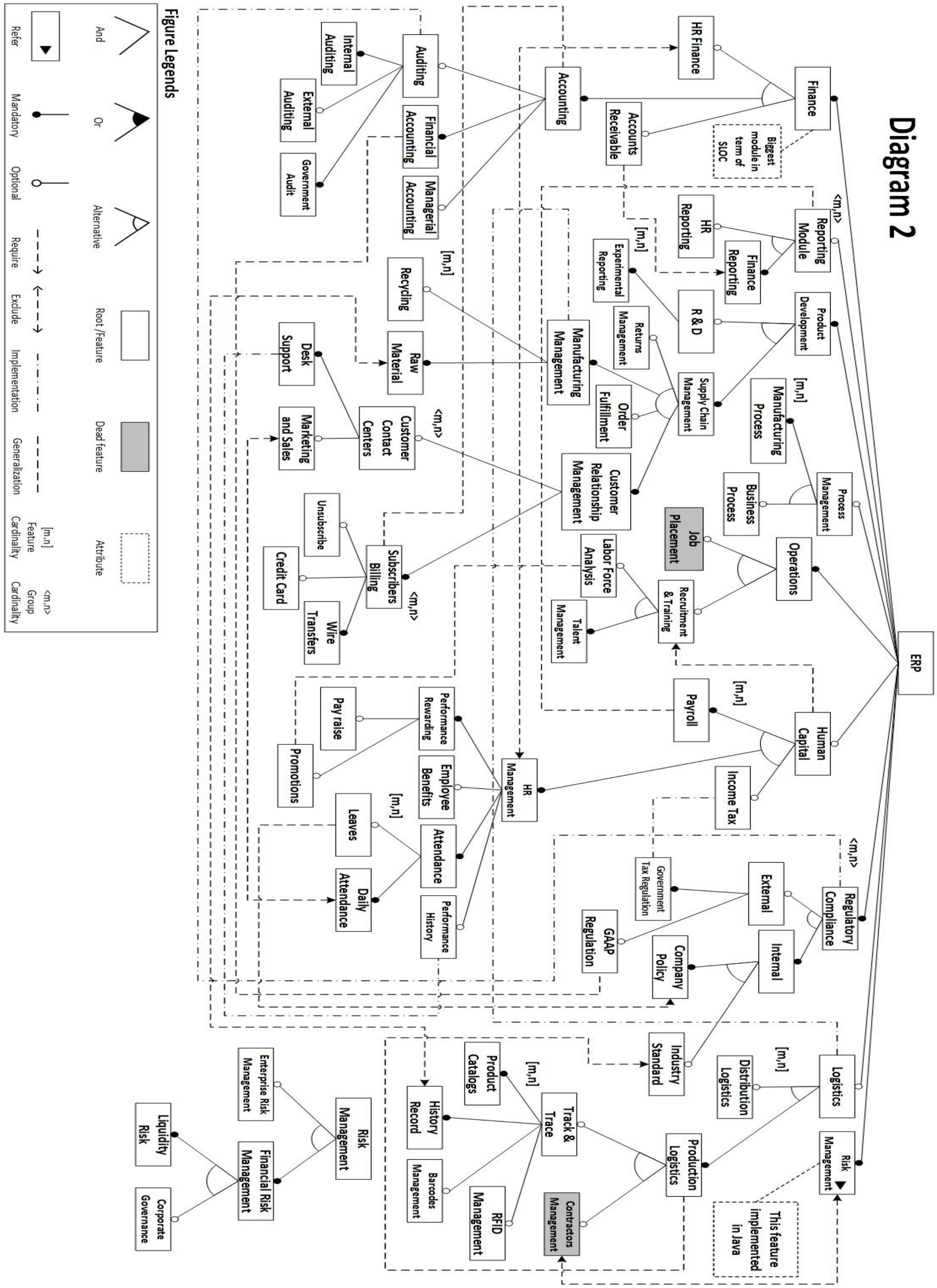
Group A, Diagram 2:



Group B, Diagram 1:



Group B, Diagram 2:



---End of the Survey---

Data for the quantitative analysis of feature diagram:

Table 10 Response time (in second)

Group A (Time in seconds)			Group B (Time is seconds)		
Subject	**Diagram 1 (ON)**	**Diagram 2 (NN)**	**Subject**	**Diagram 1 (NN)**	**Diagram 2 (ON)**
A1	2089	1508	B1	2182	2343
A2	2187	1607	B2	1943	2497
A3	2433	2143	B3	2200	2150
A4	2891	1947	B4	1655	1996
A5	2228	3052	B5	1856	1950
A6	4700	3010	B6	1878	2249
A7	2718	1896	B7	1572	2130
A8	2221	1755	B8	926	1215
A9	2922	1676	B9	2130	2385
A10	2400	1500	B10	3440	2360
A11	3610	2491	B11	1770	1847
A12	4807	2557	B12	1778	2829
A13	3306	1707	B13	1887	1935
A14	1964	1526	B14	1755	1894
A15	2236	2041	B15	1493	2061
			B16	2486	1667
			B17	1787	3152

Table 11 Errors committed in Group A.

Group A						
	Diagram 1			*Diagram 2*		
Subject	**Wrong Identification**	**Overlooked**	**Total**	**Wrong Identification**	**Overlooked**	**Total**
A1	5	0	5	3	5	8
A2	6	0	6	3	1	4
A3	8	4	12	1	0	1
A4	6	0	6	1	0	1
A5	3	0	3	3	0	3
A6	9	4	13	5	1	6
A7	4	2	6	0	0	0
A8	3	1	4	1	0	1
A9	3	0	3	1	1	2
A10	144	3	147	15	3	18
A11	4	0	4	2	0	2
A12	4	0	4	2	0	2
A13	77	2	79	61	5	66
A14	2	1	3	0	0	0
A15	5	0	5	1	0	1
				Denotes Colored Diagram		

Table 12 Errors committed in Group B.

Group B						
	Diagram 1			*Diagram 2*		
Subject	**Wrong Identification**	**Overlooked**	**Total**	**Wrong Identification**	**Overlooked**	**Total**
B1	3	0	3	7	0	7
B2	2	1	3	3	2	5
B3	4	0	4	0	0	0
B4	2	0	2	4	1	5
B5	6	0	6	4	0	4
B6	3	0	3	8	4	12
B7	2	0	2	9	4	13
B8	3	1	4	10	1	11
B9	1	0	1	1	0	1
B10	2	0	2	6	2	8
B11	1	0	1	6	1	7
B12	6	1	7	13	0	13
B13	1	0	1	4	0	4
B14	6	0	6	8	0	8
B15	4	0	4	4	1	5
B16	3	2	5	10	3	13
	Denoted Colored diagram					

Table 13 Comments & feedback Group A (Diagram 2 is New)

Q1- Which notation did you like the most (Original or New)?		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?
A1	New Notation	it was easier, specially in the relationships	divide the diagrams into smaller diagrams so it will be easier to know the relations
A2	New Notation	easier to spot relations	make relations more clear by different colors and shapes
A3	Nothing	Nothing	Nothing
A4	New Notation	I liked the New because it was simpler and clearer. The original was very exhausting unlike the New.	By making the implementation arrow filled with black color
A5	New Notation	faster to find the type of relations, features	Nothing
A6	Dead feature (maybe he says New)	It was easy to spot	Improve other notation, in the original it is waste of time and effort, my eyes hurts me because of generalization and implementation
A7	New Notation	easier, make sense & more time efficient, Dead attribute and group cardinality they are too fast	Group cardinality cloud to be of a more light color fill instead of the gray/beige color combination (it's the printing)
A8	New Notation	it took less time	Nothing
A9	New Notation	1- identify features was easy. 2- identify relations was easy. 3- I like how its super easy to identify the group cardinality	1- give generalization & implementation 2 different colors. 2 give contrasting colors for the figures. Blue & green are bad)

Q1- Which notation did you like the most (Original or New)?		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?
A10	New Notation	its better because you can separate stuff quickly and easily	maybe by using other shapes like triangles and circles
A11	New Notation	easier than the first one, New arrows was very helpful because it is hard to track the arrows	implementation arrows and generalization arrows by filling black the generalization arrow
A12	New Notation	I am not sure if the reason is because I did it after I got experience from the original one. But for sure it was easier to track the relationships lines. The different color of mandatory didn't help significantly	some relationship lines can be shorter that the way its. The yellow color is not a good choose for the optional features.
A13	New Notation	Easier to read and understand	Nothing
A14	New Notation	Easier to read, I almost exploded when I finish the first one, while the second one was somehow enjoyable	if the implementation and generalization relations had different colors
A15	New Notation	The links between the nodes were much easier to categorize, also the group and feature cardinality. I spend most the time searching for the nodes	searching for the node was not easy, I think there is no way to make searching easier otherwise everything Is perfect

Table 14 Comments & feedback Group A (Diagram 2 is New).

Q1- Which notation did you like the most (Original or New)?		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?
B1	New Notation	I choose it because is easy to navigate throw the diagram also its cleaner than the other. I really enjoy when I do the New one.	its almost perfect no need to change or add something
B2	non, but New is cleaver and cheerful	Group cardinality was obvious, but still tedious to trace	nothing
B3	I prefer New as it shows the relations easy	Because it shows the relations directly, first it confusing but with time it would be faster to extract	Choosing a color other than yellow for optional, maybe green. For optional and yellow for required because I feel green is more comfortable for the eye and its not distracting
B4	New Notation	Its easy to differentiate between groups and relations	by giving the cardinalities (mandatory & optional) different colors other than red and yellow
B5	Original	the use of arrow and colors is confusing	nothing
B6	nothing	nothing	nothing
B7	New Notation	The lines are easier to be tracked, mandatory and optional feature can be differentiated faster	it has to be organized better to be easier to be traced
B8	New Notation	The dead New notation, black color is unique among other colors, not much experience	nothing
B9	Original	The readability of the original is better in mandatory and optional, but its bad when	Just use color the links + attributes using to much icons distract me, group

	Q1- Which notation did you like the most (Original or New)?	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?
		tracing dependencies (generalization, implementation, etc.)	cardinality background color are welcome
B10	nothing	nothing	nothing
B11	Original	The original is not distracting as the New one, it has many colors	by changing the generalization to …….. And exclude to xxxxxx and drawing a circle around the group cardinality
B12	New Notation	mush faster find the node you are looking for a lot easier to understand	Distinguish between green and blue colors in low lighted. Might be hard a little bit
B13	New Notation	you can know the kind of the relation by just looking and without tracing.	Make the implementation and generalization with different colors
B14	nothing	nothing	nothing
B15	New Notation	easier to read, less confusing the new relations lines save a lot of time	Changing the yellow color to green
B16	Original	Easier to the eyes	links should be different
B17	New Notation	faster, easier to read	it might be better if there's a table with features and addresses that make searching easier. The addresses should be also in the diagram

References

[1] D. Moody, “The ‘Physics’ of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering,” *IEEE Trans. Softw. Eng.*, vol. 35, no. 6, pp. 756–779, Nov. 2009.

[2] 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.

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

[4] 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.

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

[6] D. L. Moody, P. Heymans, and R. Matulevicius, “Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of i* Visual Syntax,” 2009, pp. 171–180.

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

[8] R. S. Goonetilleke, H. Martins Shih, H. Kai On, and J. Fritsch, “Effects of training and representational characteristics in icon design,” *Int. J. Hum.-Comput. Stud.*, vol. 55, no. 5, pp. 741–760, 2001.

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

[10] 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, pp. 81–97, 1956.

[11] D. Benavides, S. Segura, and A. Ruiz-Cortés, “Automated analysis of feature models 20 years later: A literature review,” *Inf. Syst.*, vol. 35, no. 6, pp. 615–636, Sep. 2010.

[12] . Pohl , G. B c kle, and F. van der Linden, *Software product line engineering foundations, principles, and techniques*. New York, NY: Springer, 2005.

[13] M. M. Tseng and J. Jiao, “Mass Customization,” in *Handbook of Industrial Engineering*, G. S. P. D. essor founding editor founding chair member scientist H. D. Fellow, Ed. John Wiley & Sons, Inc., 2001, pp. 684–709.

[14] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, “ eature-orientedFdomainFanalysis (FODA) feasibility study,” DTIC Document, 1990.

[15] . Cz arnecki and U. W. Eisenecker, “Components and generative programming,” in *Software Engineering—ESEC/FSE’99*, 1999, pp. 2–19.

[16] M. L. Griss, J. Favaro, and M. d’ Alessandro, “Integrating feature modeling with the RSEB,” in *Software Reuse, 1998. Proceedings. Fifth International Conference on*, 1998, pp. 76–85.

[17] K.KC.KKang, J. Lee, and P. Donohoe, “Feature-oriented product line engineering,” *IEEE Softw.*, vol. 19, no. 4, pp. 58–65, 2002.

[18] D. Benavides, P. Trinidad, and A. Ruiz-Cortés, “Automated Reasoning onFFeature Models,” in *Advanced Information Systems Engineering*, O. Pastor and J. F. e Cunha, Eds. Springer Berlin Heidelberg, 2005, pp. 491–503.
[19] . Cz arnecki and S. Helsen, “Feature-based Survey of Model Transformation Approaches,” *IBM*, vol. 45, no. 3, pp. 621– 645, 2006.
[20] . Cz arnecki, S. Helsen, and U. Eisenecker, “Formalizing cardinality-based feature models and their specialization,” *Softw. Process Improv. Pract.*, vol. 10, no. 1, pp. 7–29, Jan. 2005.
[21] P. Schobbens, P. Heymans, and J.-C. Trigaux, “Feature diagrams: A survey and a formal semantics,” in *Requirements Engineering, 14th IEEE international conference*, 2006, pp. 139–148.
[22] T. von der Maßen and H. Lichter, “Deficiencies in feature models,” in *workshop on software variability management for product derivation-towards tool support*, 2004.
[23] H. Mouratidis and P. Giorgini, “Secure tropos: a security-oriented extension of the tropos methodology,” *Int. J. Softw. Eng. Knowl. Eng.*, vol. 17, no. 02, pp. 285–309, 2007.
[24] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, “Tropos: An agent-oriented software development methodology,” *Auton. Agents Multi-Agent Syst.*, vol. 8, no. 3, pp. 203–236, 2004.
[25] A. Susi, A. Perini, J. Mylopoulos, and P. Giorgini, “The Tropos Metamodel and its Use.,” *Inform. 03505596*, vol. 29, no. 4, 2005.
[26] J. C. Thomas, J. Diament, J. Martino, and R. . B ellamy, “Using the ‘Physics’ of notations to analyze a visual representation of business decision modeling,” in *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on*, 2012, pp. 41–44.
[27] OMG, “Unified Modeling Language, Version 2.4.1. Object Managment Group, Inc.,” 2012. .
[28] Object Management Group, “BPMN 2.0,” 13-Jan-2014. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/. [Accessed: 13-Jan-2014].
[29] 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.
[30] N. Genon, D. Amyot, and P. Heymans, “Analysing the Cognitive Effectiveness of the UCM Visual Notation,” in *System Analysis and Modeling: About Models*, F. A. Kraemer and P. Herrmann, Eds. Springer Berlin Heidelberg, 2011, pp. 221–240.
[31] R. J. A. Buhr and R. S. O. Casselman, *Use case maps for object-oriented systems*. Upper Saddle River, N.J.: Prentice Hall, 1996.
[32] M. Petre, “Why looking isn’t always seeing: readership skills and graphical programming,” *Commun. ACM*, vol. 38, no. 6, pp. 33–44, 1995.
[33] T. Possompès, C. Dony, M. Huchard, and C. Tibermacine, “Design of a UML profile for feature diagrams and its tooling implementation,” in *Proceedings of the Twenty-Third International Conference on Software Engineering & Knowledge Engineering*, Miami Beach, Florida, États-Unis, 2011, pp. 693–698.
[34] 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.

[35] 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.
[36] S. Siegel and N. Castellan, "Non Parametric Statistics for the Behavioral Sciences," 1988.
[37] E. Lehmann and H. D'Abrera, *Nonparametrics: Statistical methods based on ranks*. Prentice Hall (Upper Saddle River, NJ), 1998.

Vitae

Name: Mazin Ali Ahmed Saeed

Nationality: Sudanese

Date of Birth: Feb 9th, 1985

Email: mazin911@gmail.com

Addres: P.O. Box 56661, Olaya, Riyadh 11554. Kingdom of Saudi Arabia, Riyadh

Academic Background: B.Sc in Information Technology, University of Khartoum, Sudan. 2003-2007

Publications: Saeed, M. et al, 2014. Evaluating the Cognitive Effectiveness of the Visual Syntax of Feature Diagrams, Requirements Engineering, Communications in Computer and Information Science. Springer Berlin Heidelberg, pp. 180–194.

Work Experience:
Saudi Telecom Company. Riyadh 2/2012- 9/2012
Operation Engineer

Chamber of Zakat, Department of Health care (ilajj). Khartoum 12/2007 – 6/2008
Developer