

A Study Into Usability of Collaborative Software Designing on a Smartboard

- Comparing Sketching And Geometrical Modeling

Master's thesis in Software Engineering

Sinan Abdulwahhab
Yacoub Sattar

MASTER'S THESIS 2017

A Study Into Usability of Collaborative Software Designing on a Smartboard

- Comparing Sketching And Geometrical Modeling

Sinan Abdulwahhab
Yacoub Sattar



Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2017

A Study Into Usability of Collaborative Software Designing on a Smartboard
- Comparing Sketching And Geometrical
Sinan Abdulwahhab
Yacoub Sattar

© Sinan Abdulwahhab, 2017.
© Yacoub Sattar, 2017.

Supervisor: Michel Chaudron, Department of Computer Science and Engineering
Examiner: Eric Knauss, Department of Computer Science and Engineering

Master's Thesis 2017
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: UI of OctoUML modelling tool.

Typeset in L^AT_EX
Gothenburg, Sweden 2017

A Study Into Usability of Collaborative Software Designing on a Smartboard
- Comparing Sketching And Geometrical Modeling

Sinan Abdulwahhab

Yacoub Sattar

Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

This paper conducted an observational study to understand the impact which different visual appearances, sketching and geometrical figures (traditional modeling), have on the design process. The study also investigated what features are needed to make the usability of tools which combine visual appearances more intuitive for the users. OctoUML was chosen as a representation for tools which combine the use of sketching and the traditional modeling. An evaluation of the different usability characteristics was made on OctoUML. Moreover, guidelines for creation and user experience design of tools which aim to combine sketching and traditional modeling have been presented.

Keywords: UML, Sketching, Geometrical figure, Smartboard, Sketch transformation, Observational Study, Usability evaluation.

Acknowledgements

We would like to graciously thank, Michael Chaudron and Rodi Jolak. For their continuous support throughout this research study.

Sinan Abdulwahhab, Gothenburg, May 2017

Yacoub Sattar, Gothenburg, May 2017

Contents

List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Sketching feature and geometrical figures feature	1
1.2 Purpose of the study	2
1.3 OctoUML	2
2 Literature review	5
2.1 Usability	5
2.2 Usability and UML	6
2.3 UML tools	7
2.4 Improve Usability	8
2.5 Summary of Literature Review	9
3 Methods	11
3.1 Observational study session units	11
3.2 Observational study session material	12
3.3 Design	12
3.3.1 Tasks	12
3.3.2 Procedure	12
3.4 Measuring usability characteristics	13
3.5 Data Analysis	14
4 Results	15
4.1 The Effect of Visual Appearances On the Design Process (RQ1)	15
4.1.1 Qualitative data	15
4.1.2 Effectiveness and Efficiency	15
4.1.3 Satisfaction and SUS	26
4.1.3.1 System Usability Scale (SUS)	26
4.1.4 The Effect Visual Appearances Have On the Design Process (RQ1) Summary	32
4.2 Features Users Would Like To See (RQ2)	33
4.2.1 Inconvenient Design issues and future suggestions	33
4.2.2 The researchers' observations	35
4.2.3 Suggested improvements	38

5	Discussion	41
5.1	Discussing the Effect of Visual Appearances On the Design Process (RQ1)	41
5.1.1	Effectiveness and Efficiency	41
5.1.2	Understanding the Usability findings From the Observational Study findings	42
5.1.3	Satisfaction and SUS	43
5.2	What the Users Would Like to See in a Tool Which Combine the Use of Two Different Visual Appearances (RQ).	44
5.2.1	Having a variety of notations [Geometrical figures, Sketching]	44
5.2.2	Editing geometrical figures [Geometrical figures]	44
5.2.3	Selecting objects [Geometrical, Sketching]	44
5.2.4	Having dynamic relationship arrows.	45
5.2.5	Gestures [Geometrical figures, Sketching]	45
5.2.6	Undo and redo [General]	46
5.2.7	Saving the work [General]	46
5.2.8	Sketch recognition & Sketch transformation [Sketching]	46
5.2.9	A Sketch and geometrical figure view [Geometrical figures, Sketching]	47
6	Validity Threats	49
7	Conclusions & Future Work	51
	Bibliography	53
A	Appendix 1	I
B	Appendix 2	XV
B.1	Observational Study Session Tasks	XV
B.1.1	Design task 1:	XV
B.1.2	Design task 2:	XV
B.1.3	Analysis task 1:	XVI
B.2	Analysis task 2:	XVI
B.3	Qualitative Questions	XVII
B.4	Quantitative Questions.	XVII
B.4.1	User satisfaction Question	XVII
B.4.2	System Usability Scale	XVII

List of Figures

1.1	User Interface of OctoUML.	3
4.1	Efficiency and Effectiveness for the groups which solved analysis tasks	23
4.2	Efficiency and Effectiveness for the groups which solved Design tasks	23
4.3	Efficiency and Effectiveness for the groups which solved tasks using the geometrical figures feature	24
4.4	Efficiency and Effectiveness for the groups which solved tasks using the sketching feature	25
4.5	Result for the question: <i>On a scale of 1-5 how satisfied are you with the tool in general? (1 is lowest, 5 is highest).</i>	26
4.6	The result for the System Usability Scale questions for the tool where 5 is very positive.	27
4.7	The result for the System Usability Scale questions for the tool where 1 is very positive.	28
4.8	The result for the System Usability Scale questions for the sketching feature where 5 is very positive.	29
4.9	The result for the System Usability Scale questions for the sketching feature where 1 is very positive.	30
4.10	The result for the System Usability Scale questions for the geometrical figures feature where 5 is very positive.	31
4.11	The result for the System Usability Scale questions for the geometrical figures feature where 1 is very positive.	32
A.1	Edit window issue, User can tab from title to attributes but not from attributes to operators.	I
A.2	The users dragged many classes because they forgot to untoggle the selected classes.	II
A.3	Relationships arrows overlapping issue.	II
A.4	Users are having issues remembering/Choosing the right direction for relationship arrows, the same goes for the multiplicities.	III
A.5	The relationship arrow is overlapping a class.	III
A.6	The Edit window is cropped.	IV
A.7	The toggling issue in the sketching feature, the tool selects strokes by accident.	IV
A.8	Delete issue, the tool deleted strokes which were not selected.	V
A.9	Undo function did not work properly.	V
A.10	Classes overlap sketched text.	VI

A.11	Sketched text overlaps classes after dragging it.	VI
A.12	The users had to convert the sketched class to a geometrical object directly after sketching it.	VII
A.13	The users tried to delete the sketched text and it deleted a class, then they tried to undo to return the deleted class and it turned another class to sketch. Then they used the redo button and it returned the second sketched class to a geometrical class.	VII
A.14	Curved relationship arrows are not recognized.	VIII
A.15	The users would like to close the edit windows by pressing the Enter button on the keyboard.	VIII
A.16	The moving screen feature did not function properly. The users had to zoom out to have more space.	IX
A.17	The converting features does not recognize a class if it was made using more than one stroke.	IX
A.18	A class was placed into the screen but it overlapped the edit window.	X
A.19	A relationship arrow can overlap the relationship name of another relationship arrow.	X
A.20	The user tried to select the direction for the relationship arrow but it took three attempts for the tool to register.	XI
A.21	Double tapping on combined relationship arrows will open two Edit window.	XI
A.22	The user can not select a relationship arrow to remove it.	XII
A.23	The user tried to move a sketched text around but the moving is not as consistent as moving geometrical objects.	XII
A.24	The sketched strokes overlap classes and edit windows.	XIII
A.25	The users did not like the current design used to remove the relationship arrows.	XIII
A.26	The user tried to use the undo function but it did not work as it should.	XIV
A.27	A saved screenshot of a diagram obtained from the tool, some of the classes are cut and it is hard to read the information.	XIV

List of Tables

2.1	Summary of the connections between the two standards.	6
3.1	The design of the observational study sessions.	13
4.1	Result of effectiveness and efficiency	16
4.2	Effectiveness for each group design task	17
4.3	Effectiveness for each group analysis task	18
4.4	Effectiveness for each group tasks	19
4.5	Efficiency for each group design task	20
4.6	Efficiency for each group analysis task	21
4.7	Efficiency for each group task	22

1

Introduction

Software projects have evolved to bigger and more complex systems. These systems are built upon a design which is essential for the success of a mid-large system development. Sketching and creating a design is a critical task for a development process due to its nature of providing a blueprint for the software components. Many sketching and modeling tools have emerged to assist the important task of creating a design. Those tools are interesting to the industries and they are being used. However, the usability is an important factor for such tools. Having too simplistic tools could be useless for industries. On the other hand having too complex tools could be frustrating to use. Some attempts have been made to improve the usability by developing tools that combine both sketching and traditional modeling for example SUMLOW, Calico, Flexisketch, Knight and OctoUML [9, 10, 11, 12, 20]. OctoUML is a tool which assists users to sketch and design models for a software. OctoUML aims to combine the experience of sketching on a whiteboard with the traditional modeling to create a more user friendly and efficient modeling tool. OctoUML allows the users to collaboratively sketch/edit or use geometrical figures to design on a smartboard. The tool can also convert the sketched design to UML geometrical figures. OctoUML specializes in the creation of class and sequence diagram [20]. It was observed that those smartboards, even with all the new features they bring to the table, are still being used as a traditional UML modeling tool. Rather than using the sketching feature they tend to use the preinstalled geometrical figures. This defeats the purpose of creating a new tool which tries to simplify the modeling process by introducing a sketching feature. Therefore, an observational study has been conducted to determine what features the users would like to see when using a tool which combines sketching and traditional modeling. Based on the results of the study a list of suggested guidelines which can be used when creating tools which combine the use of sketching and traditional modeling has been presented. The guidelines are one of the contributions this study brings to the academic field. Additionally, an evaluation of usability in terms of effectiveness, efficiency and satisfaction for both sketching and traditional modeling has been made to decide which one has a better usability.

1.1 Sketching feature and geometrical figures feature

The paper revolves around sketching and traditional UML modeling, but for simplicity sketching has been referred to as sketching feature and traditional UML

modeling as geometrical figures feature. There is no specific naming which has been used continuously in other papers. Therefore such naming was chosen to differentiate between the two in the clearest way throughout the paper.

1.2 Purpose of the study

The purpose of this study is to evaluate the usability of the sketching feature and the geometrical figures feature and to figure out what features users would like to see in an environment which combines the two. From the purpose of the study two research questions were derived to help the study in the investigation:

- RQ1: To what extent do different visual appearances (i.e. sketching feature and geometrical figures feature) affect the design process?
- RQ2: What features users would like to see when using sketching or geometrical figures to design a UML model?

To answer these research questions an observational study has been made to observe the behavior of the participants when using the features. Afterwards a comparison between the two features have been made to evaluate which visual appearance has a better usability in terms of efficiency, effectiveness and satisfaction. The result of the study helped understanding the usability of these features and provided guidelines which should help in creating enhanced software design tools.

1.3 OctoUML

OctoUML is a UML Computer Assisted Software Engineering (CASE) tool created with the intention of allowing users to use UML geometrical figures and sketch UML notation or sketch notes collaboratively [20]. While using the tool users are allowed to create a class diagram either by dragging geometrical figures from the geometrical figures list or by transforming sketches to geometrical figures [20]. The library used for recognizing drawings is Paleo Sketch. The application is built to provide a better solution for the software designers and support them to create an architecture for their system from the idea and brainstorming phase to the documentation phase.

OctoUML is designed to be utilized the most on a touch screen device. Currently it is being used on a smartboard 800 which is a 90 inches screen. Figure 1.1 refers to a screenshot of the UI of OctoUML.

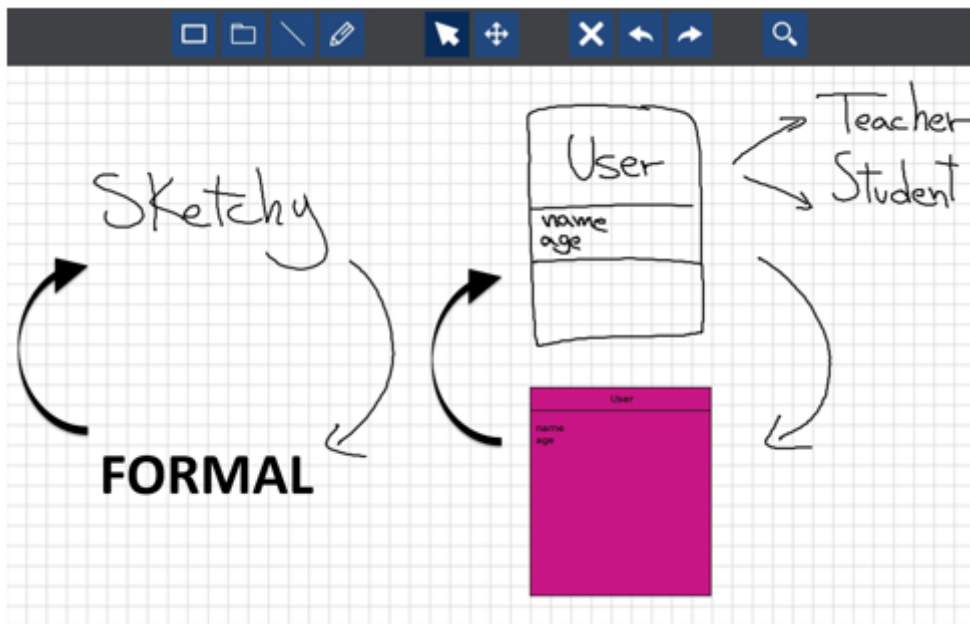


Figure 1.1: User Interface of OctoUML.

There are CASE tools in the market which combine the sketching feature and the geometrical figures feature. Since OctoUML shares the same key concept of combining the sketching feature and the geometrical figures feature, also since a direct contact with the creators and owners of OctoUML was available in case support was needed. It was decided that OctoUML would serve as a general representation or a proxy for the tools which combine the two features.

2

Literature review

Within this section different definitions of usability are presented. Afterwards there is a discussion about how usability relates to UML and why usability studies are important. Then there is a discussion about different UML tools and how they excel. Followed by a section highlighting the different attempts made to improve the usability of UML. Finally, a conclusion of the literature review which summarize everything is presented.

2.1 Usability

According to ISO/IEC 25022 (which has replaced ISO/IEC 9126), in the field of software engineering the term ‘usability’ has a similar meaning to the term ‘quality in use’ [1]. Quality in use can be assessed by measuring how well users perform the tasks (of a product) in a realistic environment [1]. Quality in use has five characteristics (effectiveness, efficiency, satisfaction, freedom from risk and context coverage) [1]. In definition usability is similar to the quality in use but it uses only three characteristics (effectiveness, efficiency and satisfaction) [1]. The definition of usability in ISO/IEC 25022 avoids using the characteristic context coverage because they believe the term usability should be used in a broader sense and not bound to a specific context. The term quality in use has the characteristic context coverage because it needs to be used in a specific context.

According to ISO 9241-11 the definition of usability is, “The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [16]. They add the characteristic context coverage because they believe the term usability should always be used in a specific context [16].

According to the research paper “The International standards for HCI and usability” which summarized and highlighted the correlations between the international standards. The paper explained that the meaning of usability according to ISO/IEC 9126 (which is currently ISO/IEC 25022) is about the usability of the interface while in ISO 9241-11 the meaning of usability is the same with addition to more focus on performance and satisfaction due to the inclusion of the context characteristic [17]. Table 2.1 introduces a table which summarizes the connections between the two standards, it was obtained from “The International standards for HCI and usability” [17].

Another definition for usability was introduced by Jakob Nielsen which suggests that the usability is a quality attribute that assesses how easy it is to use an interface

Meaning	Standard	ISO/IEC 9126-1	ISO 9241 Parts 10 and 11
Easy to use interface		“usability”	“suitable for the task, individualisation and learning, and self-descriptive, controllable, conforms with user expectations, and error tolerant”
Performance and satisfaction		“quality in use”	“usability”

Table 2.1: Summary of the connections between the two standards.

[15]. There are five different characteristics which can decide how good the usability of a product is (learnability, efficiency, memorability, errors and satisfaction) [15]. Those characteristics are similar to the ones from the international standards but were presented differently.

The international standards introduced various measures for each characteristic [1]. The effectiveness characteristic can be measured in different ways using e.g. “task completion” or “task effectiveness” measurements which utilize the “measure user performance method” [1]. The efficiency characteristic can be measured using e.g. “time efficiency” or “relative task time” measurements which also utilize the “measure user performance method” [1]. The satisfaction characteristic could be measured using e.g. “satisfaction scale” or “satisfaction questionnaire” measurements utilizing the “questionnaire method” [1]. Different measurements for different purposes for those characteristics exist in the ISO/IEC 25022 [1].

2.2 Usability and UML

Visualizing is an important factor for understandability [2]. Around a quarter of the human brain is devoted to vision and it is considered to be the best method for gathering information [2]. The processing inside the brain is divided into two, textual language processing and visual representation processing [2]. Combining the two can lead to a faster way of understanding e.g. graphs compared to reading a paragraph which explains what the graph represents [2]. UML language is the next level of representation, instead of writing a large document to explain a system process the UML language can explain a document in a model or two [2].

The notion of using visual aid to present information instead of displaying chunks of text has led to the creation of frameworks which assist and assess visual programs e.g. The Cognitive Dimensions Framework [19]. The paper, “Usability Analysis of Visual Programming Environments: A ‘Cognitive Dimensions’ Framework”, presents the framework The Cognitive Dimensions Framework. The framework is designed as a guide for assessing a visual program, the framework covers different aspects which affect the HCI of a program e.g. notations [19]. The framework states “notations” of a software should highlight information and where the mental image of the notion should fit the representative notation [19]. The notion of visualization has led to many CASE tools to be created to allow users to use the UML in a friendlier

environment [3]. Different CASE tools with different features attract different users. However, when the tools are similar in features the users will prefer ones with better usability. For example, if the CASE tool is difficult to use, the users will leave. If the tool does not clearly state its key features or it is hard to follow, users will leave [15]. This is why conducting a usability study is important to focus on such aspects to keep the users satisfied.

2.3 UML tools

The current market offers various tools for UML modeling. Although these tools differ in various ways they also share common usability features which assist users to create UML models. The common features which are shared are also considered as a necessity for a UML tool. These features are; full support to UML, robustness, HTML documentation, exporting diagrams, round-trip engineering and data modeling integration [3, 4].

- Full support to UML: UML has different diagrams under its banner which UML tools should be able to create and manipulate [3, 4].
- Robustness: These tools should be able to handle errors which has the potential to negatively impact the users. E.g. to save the model when the tool crashes [3, 4].
- HTML documentation: The tool should be able to provide HTML documentation which will assist developers with quick reference for each part of the model [3, 4].
- Exporting diagrams: The tool has the feature to export the model into different picture format e.g. PNG and JPEG [3, 4].
- Round-trip engineering: The tool should be able to generate code (forward engineering) and create model from code (Reverse engineering) in different programming languages [3, 4].
- Data modeling integration: The tool allows integrations with other data facilities. E.g. the tool should be able to allow the exported metamodel to be imported in another data modeling tool [3, 4].

There are several tools in the market which offer the above features. However, those tools have strengths and weaknesses. A couple of those tools are Visual Paradigm, Rational Rose, Magicdraw Tool, Argo UML tool and Enterprise Architect model [3, 4]. These tools are versatile in their own regards, where some tools support different versions of UML, others can export models and support several programming languages. An example of such tool is, Argo UML which supports UML 1.4 which is not the latest version, however it supports 7 different programming languages and can export models into 4 different picture formats. Another example is Visual Paradigm which is a popular tool for UML modeling [3]. This tool allows users to create UML models, generate code from the models and export pictures. Visual Paradigm has another feature which separates itself from other UML tools and it is the feature of collaborative work [5]. The tool enables users to work simultaneously on a UML project, whether it is on different part of the project or the same part e.g. two users working on the same model through different computers

[5]. Lately collaborative development has become more used in architecture development where several parties can work on a project at the same time. Tools which support collaborative work enables architecture modeling to become more efficient. The above tools have their own strengths and weaknesses which are similar in regards to functionality and the tool in general. However, the tools differ in regards to usability, according to the paper “Usability of UML modeling Tools”, the tools are similar in functionality e.g. modeling, round-trip engineering, robustness etc. but they differ when it comes to the usability [18]. The paper gathered their data on usability by conducting a usability experiment on several UML modeling tools. The usability data which was gathered from the experiment resulted from qualitative input from the participants while the various tools. These input were mostly related to functions, e.g. an input was “Improperly functioning UNDO command causes difficulties with removal of objects (they are inactive or they are not entirely deleted)” referring to a bug reported in Visual Paradigm and ArgoUML [18]. The study also analyzed the tools by counting the steps it took for the participant via a tool to create the respective modeling task. This study was concluded by comparing the input of the participants, the steps to create a diagram and time measurement of task completion, which resulted in Visual Paradigm having the highest usability score out of the chosen UML modeling tools [18].

Other CASE tools that have been created for UML are related to sketching. The idea of sketching a model instead of using geometrical figures has attracted interest due to it being more flexible. The flexible work method resembles the whiteboard experience where the users can easily draw models instead of going through a list of geometrical figures [13].

There are many different CASE tools (not specifically UML tools) with different features however only a selected few are being used [14]. There are many factors which motivate industries to invest in using CASE tools e.g. complexity of the tool, compatibility of the tool and the effectiveness of the tool [14]. It was estimated from reports that 75% of those tools are never used, 20% are used by specific groups and only 5% are used [14].

2.4 Improve Usability

In the past creating products rich with features which satisfy the needs of the users took priority [6]. However, in the current era just creating products is not enough, matching the usability factor with the product functionality is the de facto [6]. It is not a simple task to achieve; the field of usability engineering within the human computer interaction (HCI) is creating solutions which improve the interaction between humans and computers [6]. As discussed before there are many UML tools which can solve the required tasks. However, there are new innovative tools which have been created in an attempt to improve the usability of UML. A study showed that voice recognition could be used as an input method for UML [7]. The authors claimed that UML is a good target for speech recognition [7]. However, there were no follow up studies on using speech recognition as a method for modeling UML. Another study created a profile which allows modeling GUI layouts [8]. They claimed that the created profile can facilitate the integration of GUI development

into UML based software projects [8]. The profile made should model how the GUI classes should be integrated within the system, this should help reduce the fatal design flows [8]. However, there were no follow up studies which aimed to create solutions for modeling the GUI in the architecting phase. Some tools e.g. SUMLOW, Calico, Flexisketch and Knight have emerged in an attempt to improve the usability of UML [9, 10, 11, 12]. Those tools share the same idea which is sketching UML on smartboards. Each tool has different features related to the usability, Flexisketch allows the users to sketch and use geometrical figures to designs on a touch screen device, ideally on a smartboard. It also allows multiple users to work simultaneously [11]. Users can move and manipulate their sketches and insert pictures into their sketches. Calico is similar to Flexisketch but it does not have the collaboration feature [10]. SUMLOW and Knight allow the users to also sketch on a touch screen device. However, they have an algorithm which allows the tool to recognize the figures and the text sketched on the screen and afterwards it allows the users to convert the sketch to a UML diagram [9, 12].

2.5 Summary of Literature Review

The literature review managed to obtain the definition of usability. Although different definitions were presented, all of them had the key characteristics, effectiveness, efficiency and satisfaction as part of their definition. Several papers were reviewed which highlighted the importance and the reasons for the existence of UML. The main reason behind the existence was to help the users represent and understand a system without the need of reading or writing huge chunks of text. Afterwards different tools were presented each excelled in a specific area, from supporting many languages to providing a better way to work collaboratively. Each tool provided a better usability for a specific area which was needed by different users.

Finally different attempts to further improve the usability of UML were addressed. Attempts like introducing voice recognition as a method for input or introducing GUI modeling layout which complements the architecture. Other attempts to improve the usability involved creating UML sketching tools to allow users to sketch on a smartboard or other touch screen devices and then some of them allow converting sketch to geometrical figures. This study believes that those UML sketching tools are not gaining enough attention even though it seems that they provide a very interesting improvement to the usability by providing a more user friendly environment.

3

Methods

This study would like to address those two research questions:

- RQ1: To what extent do different visual appearances (i.e. sketching feature and geometrical figures feature) affect the design process?
- RQ2: What features users would like to see when using sketching or geometrical figures to design a UML model?

To address the first research question, the study needs to evaluate the usability of the sketching feature and the geometrical figures feature used in OctoUML to see how they measure against each other. Measuring the usability characteristics for the two visual appearances will determine the effect they impose on the design process. The study further determined which visual appearance is more effective or/and efficient when dealing with analysis tasks compared to a design tasks.

As for the second research question, observations need to be made and qualitative data need to be collected to determine what the users would like to see in a tool which combines the two features. An observational study seems to be the best approach to collect the necessary data to answer the second research question.

After evaluating the usability for the two features the obtained result were compared with the conducted observational study to understand the reasons behind the result from the usability evaluation.

OctoUML tool helped obtaining the necessary data for the study because it uses the sketching feature and the geometrical figures feature. Participants solved tasks related to designing an architecture using OctoUML.

3.1 Observational study session units

The population related to the study consists of everyone who uses UML tools, it includes software engineers, computer scientists, software architects and developers who have knowledge in using UML. Convenience sampling was used to gather the sample for the study [24]. The sample consists of students within the software engineering field that have completed courses which are related to software architecture and UML. Having such courses should allow the participants to solve the tasks presented in the study session. This constraint provides more accurate data when solving the tasks presented in the study session. The sample consisted of sixteen participants, the sixteen participants were placed in a group of two thus resulting in eight groups. Having a group of two allows the participants to discuss the task and obtain a better solution also to work collaboratively on OctoUML which utilizes collaborative work.

3.2 Observational study session material

The constructed observational study session had several materials. The materials were a smartboard, OctoUML design tool, session tasks and a post task questionnaire. These materials had a great amount of impact on the result and were essential to the research. The smartboard was the environment where the design tool OctoUML ran on, thus allowed multiple users to work collaboratively on OctoUML.

3.3 Design

The tasks the participants performed involved designing an architecture for a system. There were four different tasks, two analysis and two design using the sketching feature and geometrical figures feature provided by OctoUML. Having analysis and design tasks to solve was needed to determine whether the type of the task affected the solution obtained from the two different visual appearances. Two different tasks were created, design and analysis, for each one of the visual appearances. This allowed the study session to measure the usability of each feature. The study session was concluded by a post task questionnaire where the questions revolved around the usability of sketching and geometrical figures, what the users would like to see, having the smartboard as an environment, satisfaction and System Usability Scale questionnaire (SUS) [25].

There were eight groups of participants, each group had to address two tasks. Group 1 & 2 solved two design tasks first with the sketching feature and then with the geometrical figures feature. This helped determining which visual appearance suited the users best when solving a design task. Group 3 & 4 solved two analysis tasks first with the sketching feature and then with the geometrical figures feature. This helped determining which visual appearance suited the users best when solving an analysis task. Group 5 & 6 solved a design task and an analysis task using the geometrical figures feature. And group 7 & 8 solved a design task and an analysis task using the sketching feature. This was made to obtain all possible correlations, also the data obtained from group 5, 6, 7 & 8 were used to support the claims obtained from group 1 & 2 and group 3 & 4. This design also helped determining whether the type of task affected the solution obtained from the two visual appearances. The distribution of the groups is presented in Table 3.1 which explains the design of the observational study sessions.

3.3.1 Tasks

There were four different tasks, two analysis tasks and two design tasks, and one questionnaire which were used in the study. The four tasks and the questionnaire which were used in the study are located in appendix 2.

3.3.2 Procedure

The previous design section explained the design of the observational study sessions. A group of two participants solved two tasks. Each task had 5 minutes to be read

	Design task Using geometrical figures feature	Design task Using sketching feature	Analysis task Using geometrical figures feature	Analysis task Using sketching feature
Compare between the two features to determine which visual appearance was more efficient or/and effective when solving a design task	Group 1 Group 2	Group 1 Group 2		
Compare between the two features to determine which visual appearance was more efficient or/and effective when solving an analysis task			Group 3 Group 4	Group 3 Group 4
This is made to obtain all the different possible combinations, also the data obtained from group 5, 6, 7 & 8 will also be used to support the claims obtained from group 1 & 2 and group 3 & 4.	Group 5 Group 6	Group 7 Group 8	Group 5 Group 6	Group 7 Group 8

Table 3.1: The design of the observational study sessions.

and a maximum of 30 minutes to be solved. The study session took 75 minutes with a 5 minutes break between the two tasks.

The participants were handed a printed copy of the tasks and had 5 minutes to read through the task before the study session timer started. The study session was concluded with a post task questionnaire. The observational study session was recorded and the obtained solutions and questionnaire answers were stored and used in the analysis process.

A pilot test of the study session was conducted to make sure the study session would not run into unexpected circumstances. The participants in the pilot test were two students from the software engineering program who have completed software architecture courses. This made sure that the participants from the pilot test were on a similar level as the participants which participated in the study session. The pilot testers participated in solving the four design and analysis tasks and use the two features to make sure everything were functioning as it should.

3.4 Measuring usability characteristics

This study aimed to evaluate the usability of a tool which combines sketching and geometrical figures. Thus, The ISO 9241-11 definition of usability was followed because it evaluates the usability in a specific context.

To determine which feature has a better usability a measurement of the usability characteristics was needed. Measurements of effectiveness, efficiency and satisfaction were made for each visual appearance. The measurement was determined by using the guidelines presented in the ISO standard [1]. The observational study provided the necessary data to be used to calculate the characteristics which are needed to evaluate the usability of the tool.

Measuring the effectiveness. There were three mathematical formulas provided in the ISO standard for measuring the effectiveness. The closest formula which was adaptable for this study is:

$$Ev = Tc/Ta$$

Where Ev = Effectiveness, Tc = Number of tasks completed and Ta = Total number of tasks attempted.

However, due to the size of this study and the study session tasks being UML modeling, it was not possible to use the formula to its full extent to measure the effectiveness. Therefore, the measurement of the effectiveness was based on how correct the obtained solution for each task was, since effectiveness is defined as “accuracy and completeness with which users achieve specified goals” [1]. The solution was evaluated upon a grading criteria and it was scored between 0-10, where 0 is lowest and 10 is highest.

Measuring the efficiency. The measurement of the efficiency was calculated using the formula:

$$Ey = Ev/T$$

Where Ey = Efficiency, Ev = Effectiveness and T = Time. The formula fits the study well however, task effectiveness was the grade of the task. Which by default changed the measurement and size of the result. However, the result was sufficient to indicate the efficiency. E.g. an efficiency result would be $X = (6/10) / 900$ seconds = 0.00077778. Having such low numbers is messy to read. Therefore, the solutions were multiplied by 10000 and rounded up to one significant figure. Thus the efficiency result will be 7.8.

Measuring the satisfaction. The satisfaction was measured by a post task questionnaire. Questions about satisfaction and SUS questions were used. The satisfaction was measured by the average score of each question, where the score came from a subjective 1-5 disagree to agree scale.

3.5 Data Analysis

The gathered data were both qualitative and quantitative. Since the quantitative questions were specific for satisfaction and SUS it was directly collected and used to measure the satisfaction characteristic. Analytic memo was used as a data analysis method for the qualitative data. The method allows researchers to theme up qualitative data and analyze the meaning of the data in a more in-depth manner [21, 22]. This allows researcher to not only analyze the data in depth and theme the data but also to reduce any assumption and subjective perspective [22]. The data for the other two sub characteristics, efficiency and effectiveness, was gathered by analyzing the recordings of the designing process from each study session.

4

Results

This observational study had 8 study sessions where data were collected. Qualitative and quantitative data were gathered to help answer the research questions. This section presents two subsections where the result related to each research question is presented.

4.1 The Effect of Visual Appearances On the Design Process (RQ1)

This subsection presents data related to the first research question. This includes related qualitative and quantitative data, as well as data which revolves around the three characteristics of the ISO standard.

4.1.1 Qualitative data

The post task questionnaire had qualitative questions regarding what feature the users believed was better and was more satisfying to use. Out of all the qualitative questions there were 2 specific questions,

1. *Which feature (Sketching or Geometrical) would you say is better?. And why do you think it is better.*
2. *When dealing with an analysis task what feature (Sketching or Geometrical) would you prefer to use?*

For the first question 15 out of 16 participants believed that the geometrical figures feature was better and more satisfactory and 1 believed the sketching feature was better. This indicates that the participants of the study believed the geometrical figures feature was better than the sketching feature.

For the second question 3 out of 4 participants who explicitly solved analysis tasks using sketching once and geometrical figures once. The participants believed that the geometrical figures feature was better when dealing with an analysis task, and one believed that the sketching feature was better. This data indicates that the participants would prefer using geometrical when dealing with analysis tasks.

4.1.2 Effectiveness and Efficiency

Results which influenced the effectiveness and the efficiency of the visual appearances will be presented. The effectiveness and the efficiency of each solved task has been calculated and presented in Table 4.1.

4. Results

Design task ----- Using geometrical figures feature	Design task ----- Using sketching feature	Analysis task ----- Using geometrical figures feature	Analysis task ----- Using sketching feature
Group 1: Time: 11:27 minutes Effectiveness: 5/10 Efficiency: 7.3 Group 2: Time: 29:21 minutes Effectiveness: 5/10 Efficiency: 2.8	Group 1: Time: 13:25 minutes Effectiveness: 6/10 Efficiency: 7.4 Group 2: Time: 29:31 minutes Effectiveness: 7.5/10 Efficiency: 4.2		
		Group 3: Time:13:30 minutes Effectiveness: 6/10 Efficiency: 7.4 Group 4: Time:11:54 minutes Effectiveness: 5/10 Efficiency: 7.0	Group 3: Time: 5:42 minutes Effectiveness: 2/10 Efficiency: 5.8 Group 4: Time: 14:19 minutes Effectiveness: 5/10 Efficiency: 5.8
Group 5: Time: 30:00 minutes Effectiveness: 9/10 Efficiency: 5 Group 6: 27:53 minutes 1673 seconds Effectiveness: 4/10 Efficiency: 2.4	Group 7: Time: 20:53 minutes Effectiveness: 6/10 Efficiency: 4.8 Group 8: Time: 28:53 minutes Effectiveness: 4/10 Efficiency: 2.3	Group 5: Time: 29:10 minutes Effectiveness: 8/10 Efficiency: 4.6 Group 6 Time: 28:13 minutes Effectiveness: 7.5/10 Efficiency: 4,4	Group 7: Time: 21:31 minutes Effectiveness: 7/10 Efficiency: 5.4 Group 8: Time: 7:58 minutes Effectiveness: 7/10 Efficiency: 13.8

Table 4.1: Result of effectiveness and efficiency

Table 4.1 presents the result of the effectiveness and efficiency number. The data allows the study to draw 4 different implication. Due to the number of the study participants this study cannot draw strong conclusions but merely display possible implications of the study's result.

Effectiveness:

- *A conclusion cannot be drawn on which feature is more effective when dealing with a design task.* Table 4.2 displays the effectiveness of each design task. The comparisons ended up non conclusive since 2 comparisons are even and approximately even. The other 2 comparison, had one with geometrical figures feature having higher effectiveness and the other had sketching with higher effectiveness.

Design task ----- Using geometrical figures feature	Design task ----- Using sketching feature	Comparison Result ----- Feature with Highest Effectiveness
Group 1: Effectiveness: 5/10	Group 1: Effectiveness: 6/10	Approximately even (No big difference in values)
Group 2: Effectiveness: 5/10	Group 2: Effectiveness: 7.5/10	Sketching feature
Group 5: Effectiveness: 9/10	Group 7: Effectiveness: 6/10	Geometrical figures feature
Group 6: Effectiveness: 4/10	Group 8: Effectiveness: 4/10	Even

Table 4.2: Effectiveness for each group design task

1. *Geometrical figures feature is more effective than the sketching feature when dealing with an analysis task.* This can be seen in Table 4.3, 3 out of 4 comparisons were even and approximately even. There was one comparison, Group 3 vs Group 3, which indicated that geometrical figures feature had higher effectiveness.

4. Results

Analysis task ----- Using geometrical figures feature	Analysis task ----- Using sketching feature	Comparison Result ----- Feature with Highest Effectiveness
Group 3: Effectiveness: 6/10	Group 3: Effectiveness: 2/10	Geometrical figures feature
Group 4: Effectiveness: 5/10	Group 4: Effectiveness: 5/10	Even
Group 5: Effectiveness: 8/10	Group 7: Effectiveness: 7/10	Approximately even (No big difference in values)
Group 6: Effectiveness: 7.5/10	Group 8: Effectiveness: 7/10	Approximately even (No big difference in values)

Table 4.3: Effectiveness for each group analysis task

- Overall, the geometrical figures feature is more effective than the sketching feature. This can be seen in Table 4.4, 5 out of 8 comparisons were even and approximately even. 2 out of 8 comparisons had geometrical figure feature with higher effectiveness and one comparison had sketching feature with higher effectiveness.

Geometrical figures feature	Sketching feature	Comparison Result ----- Feature with Highest Effectiveness
Group 1: Effectiveness: 5/10	Group 1: Effectiveness: 6/10	Approximately even (No big difference in values)
Group 2: Effectiveness: 5/10	Group 2: Effectiveness: 7.5/10	Sketching feature
Group 5: Effectiveness: 9/10	Group 7: Effectiveness: 6/10	Geometrical figures feature
Group 6: Effectiveness: 4/10	Group 8: Effectiveness: 4/10	Even
Group 3: Effectiveness: 6/10	Group 3: Effectiveness: 2/10	Geometrical figures feature
Group 4: Effectiveness: 5/10	Group 4: Effectiveness: 5/10	Even
Group 5: Time: 30:00 minutes Effectiveness: 8/10	Group 7: Time: 20:53 minutes Effectiveness: 7/10	Approximately even (No big difference in values)
Group 6: Effectiveness: 7.5/10	Group 8: Effectiveness: 7/10	Approximately even (No big difference in values)

Table 4.4: Effectiveness for each group tasks

Efficiency:

1. *Sketching is more efficient than the geometrical figures for design tasks.* This can be seen in Table 4.5, 3 out of 4 comparisons were even and approximately even. There was one comparison, Group 2 vs Group 2, which indicated that sketching feature had higher efficiency.

4. Results

<u>Design task</u> Using geometrical figures feature	<u>Design task</u> Using sketching feature	<u>Comparison Result</u> Feature with Highest Effectiveness
Group 1: Efficiency: 7.3	Group 1: Efficiency: 7.4	Approximately Even (Non significant difference in values)
Group 2: Efficiency: 2.8	Group 2: Efficiency: 4.2	Sketching feature
Group 5: Efficiency: 5	Group 7: Efficiency: 4.8	Approximately Even (Non significant difference in values)
Group 6: Efficiency: 2.4	Group 8: Efficiency: 2.3	Approximately Even (Non big difference in values)

Table 4.5: Efficiency for each group design task

2. *Geometrical figures is more efficient than the sketching feature when dealing with an analysis task.* This can be seen in Table 4.6 where 2 out of 4 comparisons implied that using geometrical figures is more efficient than sketching. One comparison implied that sketching feature was more efficient and there was one comparison which was approximately even.

<u>Analysis task</u> Using geometrical figures feature	<u>Analysis task</u> Using sketching feature	<u>Comparison Result</u> Feature with Highest Effectiveness
Group 3: Efficiency: 7.4	Group 3: Efficiency: 5.8	Geometrical Figures Feature
Group 4: Efficiency: 7.0	Group 4: Efficiency: 5.8	Geometrical Figures Feature
Group 5: Efficiency: 4.6	Group 7: Efficiency: 5.4	Approximately even (No big difference in value)
Group 6: Efficiency: 4.4	Group 8: Efficiency: 13.8	Sketching feature

Table 4.6: Efficiency for each group analysis task

- *Overall, a conclusion cannot be drawn on which feature is more efficient.* This can be seen in table 4.7, 4 out of 8 comparisons are even or approximately even. The other 4 comparisons are divided in two. Where two comparisons implied that sketching is more efficient and the other 2 implied that geometrical figures is more efficient. Thus, it not possible to draw any conclusions.

4. Results

<u>Design task</u> Using geometrical figures feature	<u>Design task</u> Using sketching feature	<u>Comparison Result</u> Feature with Highest Effectiveness
Group 1: Efficiency: 7.3	Group 1: Efficiency: 7.4	Approximately Even (Non significant difference in values)
Group 2: Efficiency: 2.8	Group 2: Efficiency: 4.2	Sketching feature
Group 5: Efficiency: 5	Group 7: Efficiency: 4.8	Approximately Even (Non significant difference in values)
Group 6: Efficiency: 2.4	Group 8: Efficiency: 2.3	Approximately Even (Non big difference in values)
Group 3: Efficiency: 7.4	Group 3: Efficiency: 5.8	Geometrical Figures Feature
Group 4: Efficiency: 7.0	Group 4: Efficiency: 5.8	Geometrical Figures Feature
Group 5: Efficiency: 4.6	Group 7: Efficiency: 5.6	Approximately Even (Non significant difference in values)
Group 6: Efficiency: 4.4	Group 8: Efficiency: 13.8	Sketching feature

Table 4.7: Efficiency for each group task

Charts 4.1, 4.2, 4.3 and 4.4 display different visualizations of the relation between effectiveness and efficiency when solving analysis tasks, design tasks, when using the geometrical figures feature or when using the sketching feature.

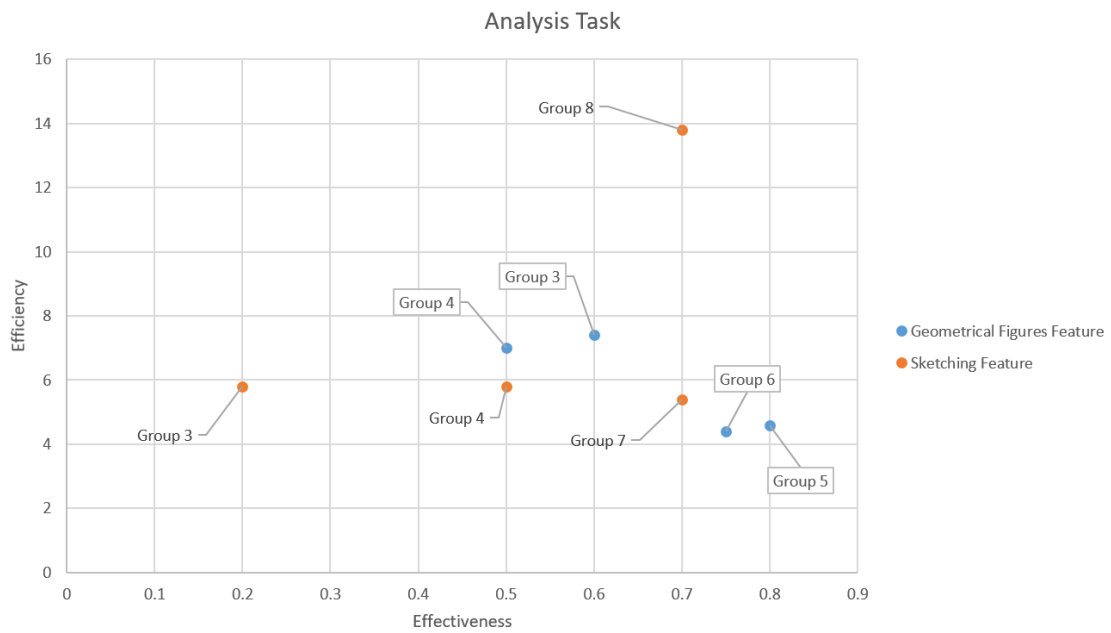


Figure 4.1: Efficiency and Effectiveness for the groups which solved analysis tasks

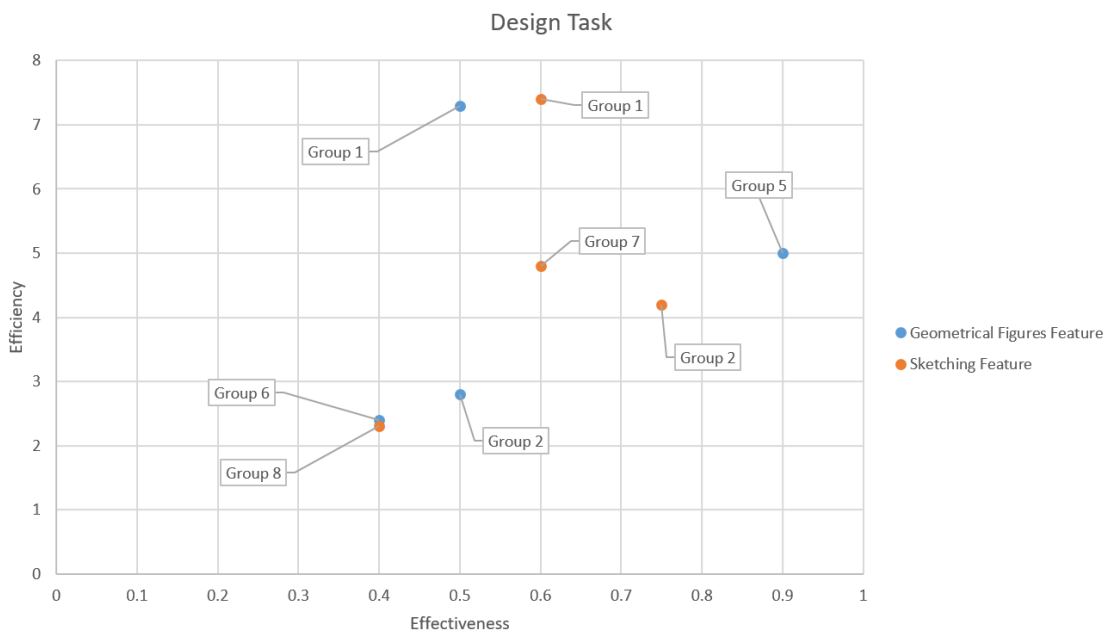


Figure 4.2: Efficiency and Effectiveness for the groups which solved Design tasks

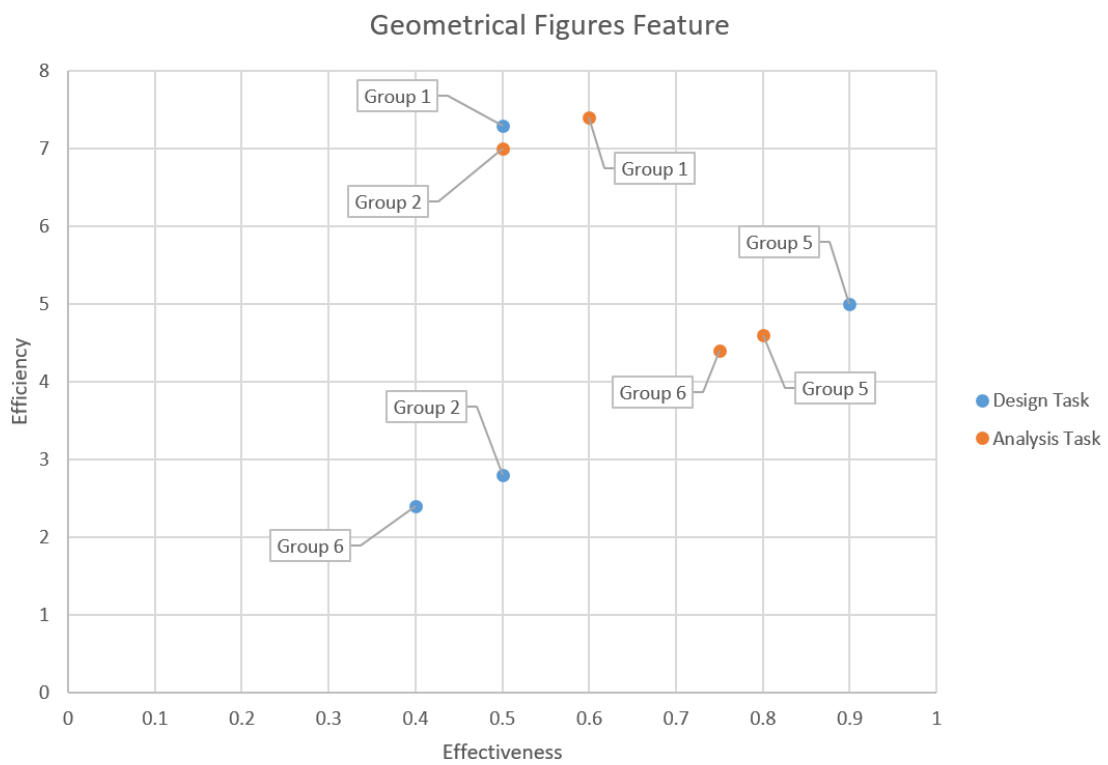


Figure 4.3: Efficiency and Effectiveness for the groups which solved tasks using the geometrical figures feature

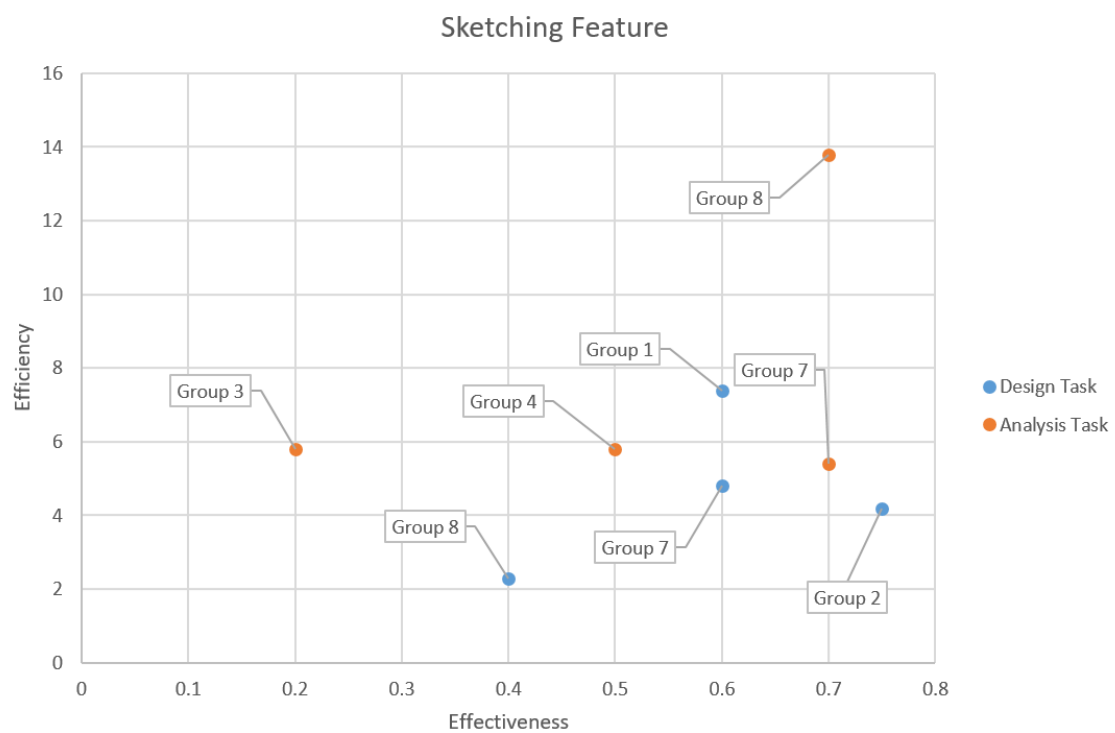


Figure 4.4: Efficiency and Effectiveness for the groups which solved tasks using the sketching feature

4.1.3 Satisfaction and SUS

To have an overall measure of the users' satisfaction with the tool a box plot which represents the questionnaires' responses have been made. A list of the quantitative questions from the questionnaire which were used in the box plot are presented below:

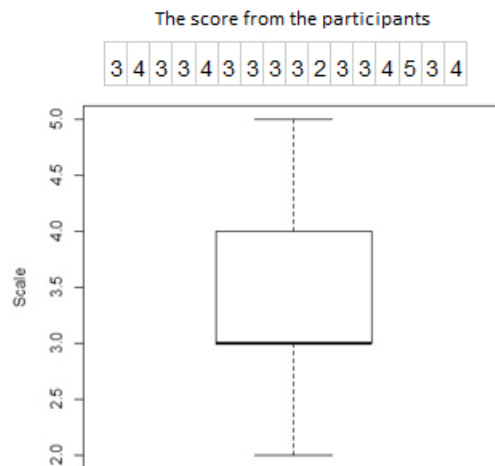


Figure 4.5: Result for the question: *On a scale of 1-5 how satisfied are you with the tool in general? (1 is lowest, 5 is highest).*

As can be seen in figure 4.5, the data points in the row show that the majority of the participants are mildly satisfied with the system because the majority of the scores are ranging from 3 to 5. However, there is one participant which gave the system a score of 2. Despite the one negative data point which gave a score of 2, it can be concluded that the majority of the participants are satisfied with the system. The satisfaction level per participant goes from modestly satisfied 3 points to extremely satisfied 5 points. The median score is 3 which implies that the consensus were modestly satisfied with the tool.

4.1.3.1 System Usability Scale (SUS)

System Usability Scale (SUS) is a standard 10 question questionnaire which evaluates usability of a tool. SUS is a scale based questionnaire where the question goes from 1-5. 1 is the lowest which indicates the users disagree and 5 is the highest which indicates the users agree. A list of the SUS questions will be presented and 3 different box plots of the users' responses will also be presented.

Q1- I think that I would like to use this system frequently.

Q2- I found the system unnecessarily complex.

Q3- I thought the system was easy to use.

Q4- I think that I would need the support of a technical person to be able to use this system.

Q5- I found the various functions in this system were well integrated.

Q6- I thought there was too much inconsistency in this system.

Q7- I would imagine that most people would learn to use this system very quickly.

Q8- I found the system very cumbersome to use.

Q9- I felt very confident using the system.

Q10- I needed to learn a lot of things before I could get going with this system.

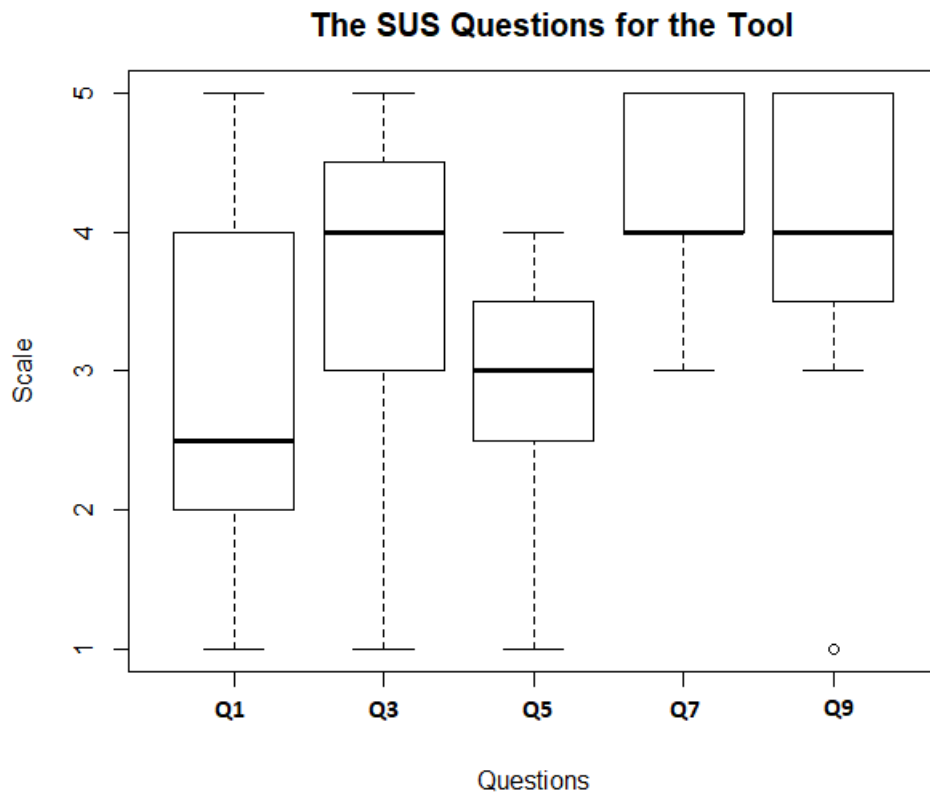


Figure 4.6: The result for the System Usability Scale questions for the tool where 5 is very positive.

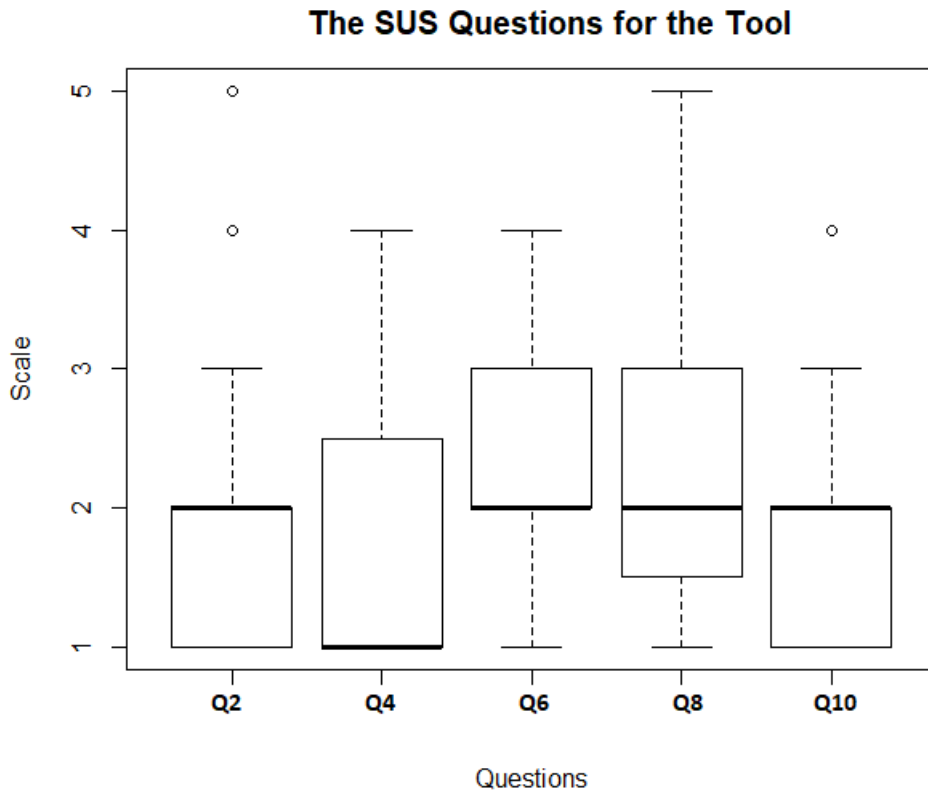


Figure 4.7: The result for the System Usability Scale questions for the tool where 1 is very positive.

Figure 4.6 displays the questions where 5 is very positive and Figure 4.7 displays the questions where 1 is very positive. The consensus of the two figures implies that the participants find the tool usable. However, Q5, "I found various functions in this system well integrated", produces a median value of 3 and lean towards the unsatisfied category. This could be because the tool has some design flaws and bugs. As for Q1, "I think that I would like to use this system frequently", which produced a median of 2.5 which is unsatisfactory, it could be because the tool does not support many notations to allow the users to use it frequently and also because it has some design flaws. Thus some users would not consider using this tool frequently.

In the case of Q2, "I found the system unnecessarily complex", 2 participants out of 16 found the system complex and gave points of 5 and 4 which ended up as outliers. However, looking at the chart the majority of participants believed that the system was not unnecessarily complex.

There are 2 more outliers one for Q9 and the other for Q10. Q9, "I felt very confident using the system", there was one participant who did not feel confident using the system, and gave it a 4.

As for Q10, "I needed to learn a lot of things before I could get going with this system", there was one participant who gave the question a 4. Despite the few outliers and Q1 and Q5, the consensus of each question leans towards the satisfied category.

Figure 4.6 and 4.7 displayed the data from all of the participants on their view of the tool. However, the differences in data might be affected by the feature they used. Figures 4.8, 4.9, 4.10 and 4.11 introduce box plots of data points obtained from participants who explicitly used one feature during the study. The sections below will display the SUS data from the participants who explicitly used one feature.

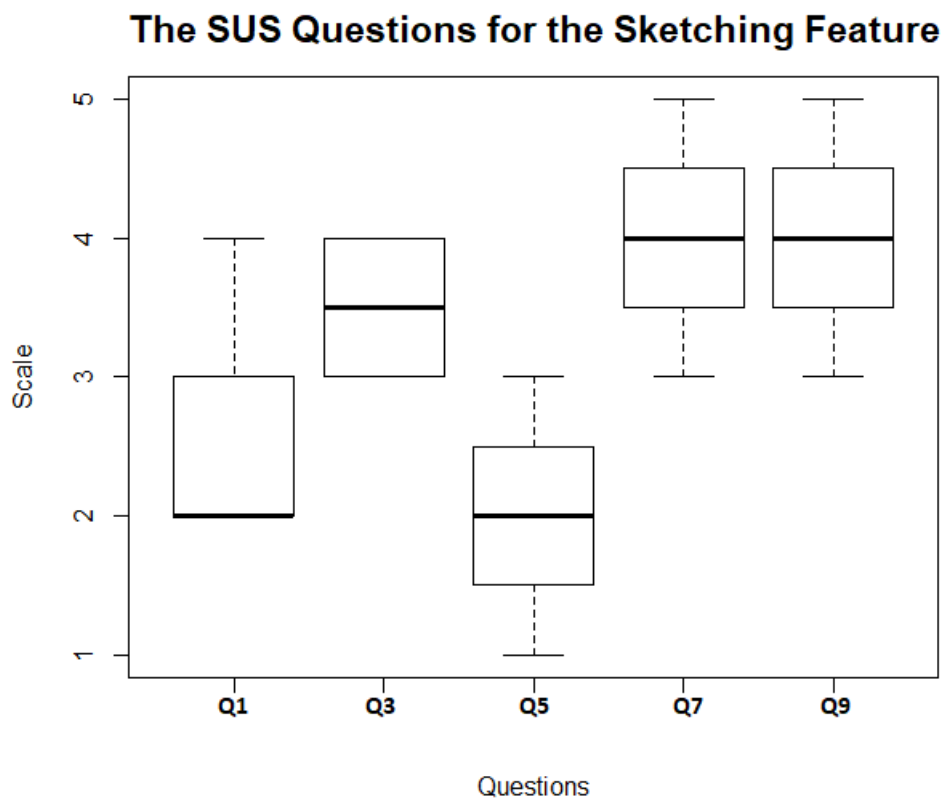


Figure 4.8: The result for the System Usability Scale questions for the sketching feature where 5 is very positive.

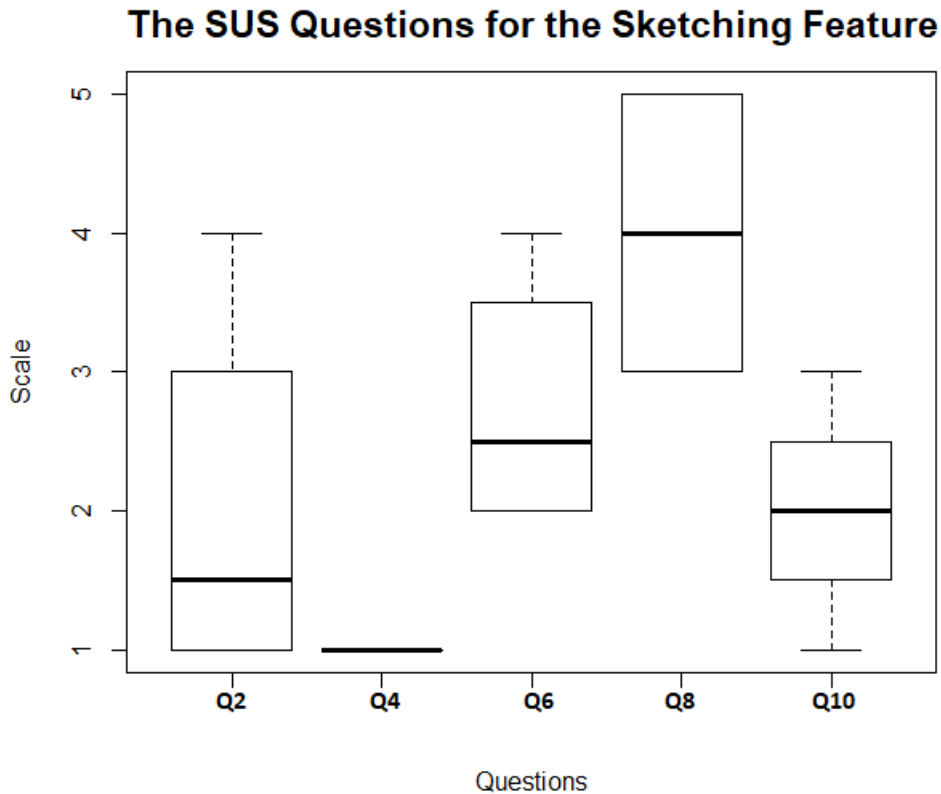


Figure 4.9: The result for the System Usability Scale questions for the sketching feature where 1 is very positive.

In Figure 4.8 and 4.9 the sketching feature indicates that the participants are agreeable on the matter of the feature being usable. There are no outliers which indicates that the participants have a consensus of each question. In the case of Q4, "I think that I would need the support of a technical person to be able to use this system", all participants agreed upon score of 1 which indicates they thought the feature was easy to learn. While using the sketching feature Q1, "I think that I would like to use this system frequently", received a median of 2 which is an unsatisfied answer. This could be because the sketching feature has many design flaws and bugs. Q5, "I found the various functions in this system were well integrated", received a median of 2, Q8 received a median of 4, these medians support the assumption made for Q1 regarding the design flaws and bugs.

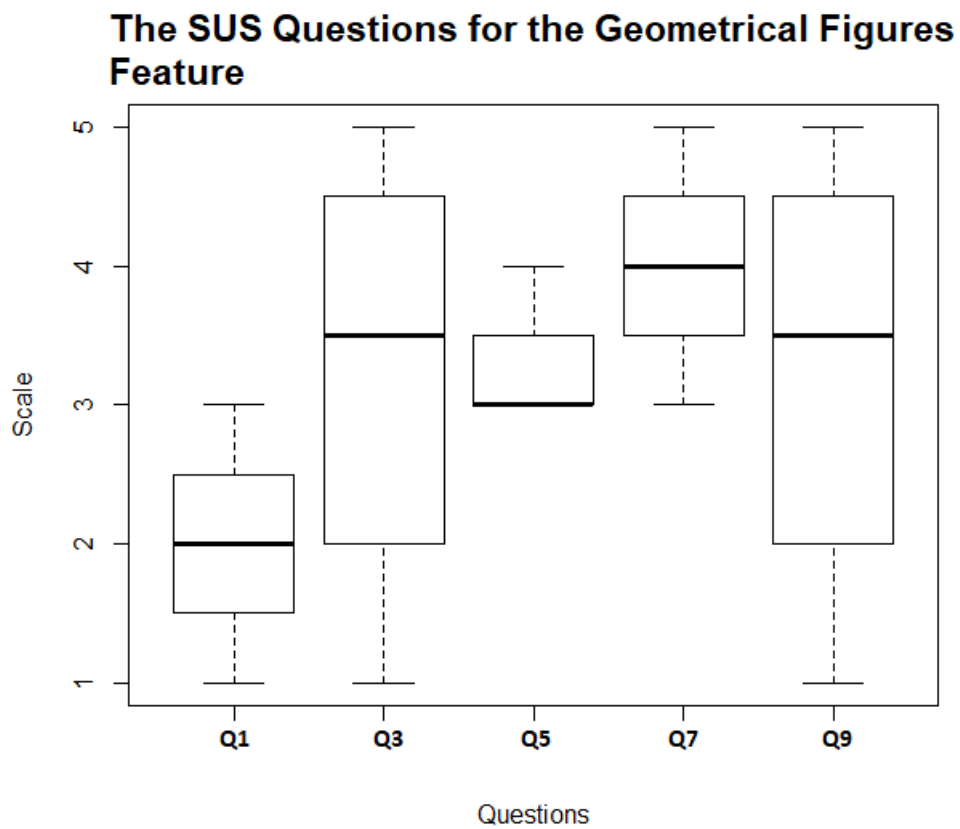


Figure 4.10: The result for the System Usability Scale questions for the geometrical figures feature where 5 is very positive.

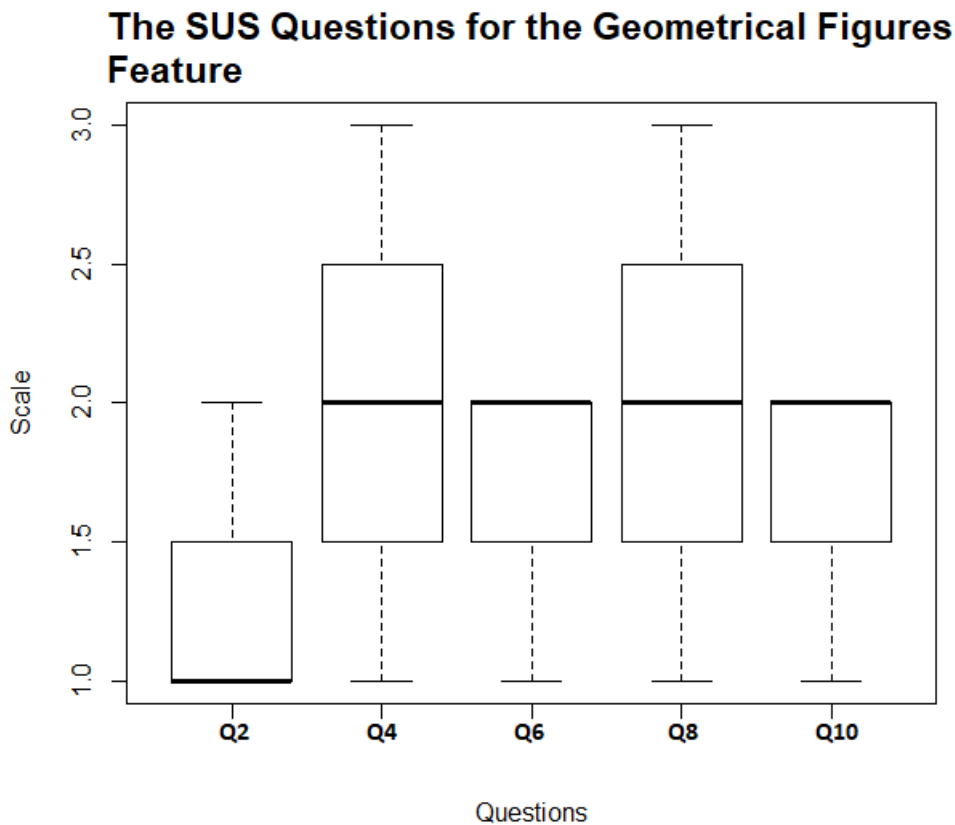


Figure 4.11: The result for the System Usability Scale questions for the geometrical figures feature where 1 is very positive.

In figure 4.10 and 4.11 the geometrical figures feature indicates that the participants are agreeable on the matter of the feature being usable and satisfiable. There are no outliers in the figures and all questions leans towards satisfactory. However, Q1, "I think that I would like to use this system frequently", produced a median of 2 which is unsatisfactory. This could be because the tool does not support many notations to allow the users to use it frequently.

4.1.4 The Effect Visual Appearances Have On the Design Process (RQ1) Summary

The above subsections present the results which were acquired during the research, which answers the first research question, "To what extent do different visual appearances (i.e. sketching feature and geometrical figures feature) affect the design process?"

From the presented data above it can be concluded that visual appearances indeed affect the design process. The subsection effectiveness and efficiency present 4 possible implications.

Effectiveness:

1. *Geometrical figures feature is more effective than the sketching feature when dealing with an analysis task.*

2. Overall, the geometrical figures feature is more effective than the sketching feature.

Efficiency:

1. Sketching is more efficient than the geometrical figures for design tasks.
2. Geometrical figures is more efficient than the sketching feature when dealing with an analysis task.

These possible implications imply that visual appearances and different type of tasks indeed affect design process. The visual appearance does not only affect the design process but also the user experience. This can be seen by the different implications and measurements acquired from each task. The subsection, Qualitative data, presents users' opinion which implied that most of the participants preferred geometrical figures feature over the sketching feature. The box plots for the SUS questions showed mostly satisfactory medians with some questions received unsatisfactory median. The usability evaluation could not accurately determine the reasons for the unsatisfactory medians by itself. However, the observational data and the in depth qualitative data will explain in details in the next section.

4.2 Features Users Would Like To See (RQ2)

Within this section the data related to the users' observations and suggestions will be presented. Also, suggestions and observation noted by the researchers from the study sessions will be presented.

4.2.1 Inconvenient Design issues and future suggestions

The design issues reported by the users and the suggested solutions and recommendations for new designs will be presented.

1. Select Element(s) [Geometrical figures, Sketching]

The current design to select objects is flawed. OctoUML does not deselect in specific cases e.g. when the user decides to edit a class the other selected classes will remain selected in the background. This caused users to forget to deselect and when the users decide to move the edited class both of the classes will be moved simultaneously. Many users complained about this design issue and would like to see a new design for selecting elements.

2. Need erasing tool for sketching [Sketching]

The users do not like the current method for deleting a sketch. OctoUML's current method for deleting requires the user to select a sketch. Then OctoUML deletes one stroke at a time starting with the latest stroke. The users would like to see some sort of a physical tool which can be used for erasing.

3. Revamp the usability of the relationship arrows [Geometrical figures]

The current way OctoUML decides on the direction of the relationship arrow is confusing for the users. There are no proper indication as to which direction the

relationship arrow starts from and where it ends. A user said “Add direction indicator for the relationship arrows”. Many users found themselves guessing on the direction of the relationship arrow. The current way for deciding the direction of the relationship arrows is not intuitive.

4. The selected object color should be consistent [Geometrical figures, Sketching]

Currently OctoUML has different colors when it comes to highlighting objects. E.g. classes have red color and highlighted sketches has purple. A user said “The selection color should be red.” Indicating that the coloring should be consistent.

5. More anchors and connectors on classes [Geometrical figures]

The current design of OctoUML’s relationship arrows overlaps with each when they are connected to the same anchor/connector on a class. Which leads to improper readability of the model. A user said “*Adding more connectors to a class rather than having all of them combined together*” to solve this problem.

6. Dynamic relationships arrows [Geometrical figures]

Currently if a relationship arrow is connected to a class, the only way to change the location of that relationship arrow would be to delete it and create a new one. Also when two relationship arrows cross each other it becomes confusing. A user said “*Add a way to move relationship arrows between classes, and the path they take so they wouldn’t collide with other classes*”, as a solution for this problem.

7. More UML notation [Geometrical figures]

Currently OctoUML supports notations of class diagram and sequence diagram. The users would like to see support for more notations. A user said “*More geometrical features, might have been good. Such as components*”. And another user said “*Component and deployment diagrams sketching could be faster but the tool doesn’t recognize them*”. So it would be a good addition for OctoUML to have more notations for the geometrical figures feature and to be able to recognize sketches of such notations to also improve the sketching feature.

8. Edit window improvement [Geometrical figures]

The Edit window is inconsistent with tabbing from one textbox to another. In some cases pressing TAB can move to another textfield and in other cases it just makes a space. A user said “*Add tabbing between fields when writing*”. referring to this problem.

9. Text transformation [Sketching]

OctoUML currently recognizes only sketched classes and associations. After a user sketches a diagram OctoUML will not be able to recognize any text written or different types of relationship arrows. A user said “*It could be possible to lose a lot of work done by sketching when it can’t be converted into geometrical. Especially text*”. Referring to the limitation of the recognition feature. Another user said “*Felt easier to draw things, would be good if the tool get better text recognition.*”

supporting the claim.

4.2.2 The researchers' observations

Within this section all of the design issues and bugs observed during the study will be presented. There will be a subsection introducing a list of all of the discovered bugs and issues and another subsection introducing a list of suggested improvements.

During the analysis of the recorded data, design issues and bugs were discovered. Table 4 introduces the design issues and bugs discovered using the geometrical figures feature. Table 5 introduces the issues and bugs discovered using the sketching feature. Table 6 introduces issues represents general issues with the tool. The occurrence of the issues were registered once per recorded observation session, there were 16 recorded observation sessions, 8 tasks were solved using the geometrical figures feature and 8 tasks were solved using the sketching feature. In some cases the occurrence of an issue exceeds 8 and that is because the feature related to that issue was used during both the sketching feature and the geometrical figures feature. Issues and bugs discovered while using the geometrical figures feature:

1. Edit window	<p>Issue 1: When adding information to the classes the keyboard is being used to insert the information. After adding information in the “title” textfield pressing TAB can switch to the “attributes” textfield. However pressing TAB afterwards will not switch to the Operations textfield. This is not consistent. Refer to Figure A.1 in appendix a. Occurrence: 5.</p> <p>Issue 2: Edit node does not overlap screen or relocate itself. It results in a cropped view of the edit node window. The users have to zoom out or move the screen to add information. Refer to figure A.2 in appendix a. Occurrence: 9.</p> <p>Issue 3: Press Enter button on the keyboard to close an edit window. The users seemed to naturally press enter to close an edit window. They seemed to get frustrated when it did not follow their need. Refer to figure A.3 in appendix a. Occurrence: 5.</p>
2. Toggle	<p>The users tend to pull many classes at the same time when they are trying to move one. This is because they did not untoggle the other classes. Refer to figure A.4 in appendix a. Occurrence: 9.</p>

<p>3. Relationship arrows</p>	<p>Issue 1: If a class has many relationship arrows connected to it from the same side the connections will overlap and the relationships will not be understood. Refer to figure A.5 in the appendix a. Occurrence: 9.</p> <p>Issue 2: Users have issue with choosing/remembering the direction of the relationship also the locations of the multiplicities. Refer to figure A.6 in appendix a. Occurrence: 5.</p> <p>Issue 3: When an relationship arrow is dragged between two classes and there is a class in between the arrow will overlap the class. If the overlapped class is moved the relationship arrow will hide behind the class. Refer to figure A.7 in appendix a. Occurrence: 4.</p> <p>Issue 4: A relationship arrow can overlap the relationship name of another relationship arrow. Refer to figure A.21 in appendix a.</p> <p>Issue 5: The user tried to select the direction of the relationship arrow. The tool directly selected Start_To_End. The user tried again to change the direction and after three attempts the tool choose the right direction. Refer to figure A.22 in appendix a. Occurrence: 1.</p> <p>Issue 6: Double tapping on overlapped relationship arrows will open two Edit windows. Refer to figure A.23 in appendix a. Occurrence:2.</p> <p>Issue 7: The user tried to select a relationship arrow to remove it however, the tool did not select the relationship arrow. Refer to figure A.24 in appendix a. Occurrence: 1.</p> <p>Issue 8: The users did not like the way the tool removes the relationship arrows. Refer to figure A.27 in appendix a. Occurrence: 1.</p>
<p>4- CRUD</p>	<p>The user deleted a class by mistake when he was trying to delete a relationship arrow. The user tried to use the undo function but the undo function did not return the last deleted class. However, it returned some random empty classes. Refer to figure A.28 in appendix a. Occurrence:1.</p>

Issues and bugs discovered while using the sketching feature:

<p>1. Toggled sketch</p>	<p>Issue 1: When a user sketch the tool toggles strokes by accident. Refer to figure A.9 in appendix a. Occurrence: 5.</p>
--------------------------	---

2. CRUD	<p>Issue 1: The delete function does not work as it should. The user had many strokes toggled and the delete function did not delete all the toggled ones. Some toggled strokes were not deleted and some un-toggled strokes were deleted. Refer to figure A.9 and A.10 in appendix a. Occurrence: 5.</p> <p>Issue 2: The user selected a sketched word and a geometrical figure. The user tapped the delete button and the tool deleted a letter from the sketched word and the geometrical figure. Refer to figure A.15 in appendix a. Occurrence: 2.</p> <p>Issue 3: The undo function did not work properly. The user pressed the undo button to restore the deleted strokes but not all of the strokes were retrieved. Refer to figure A.9, A.10 and A.11 in appendix a. Occurrence: 1.</p> <p>Issue 4: The undo function did not work properly when used in a diagram containing geometrical figures and sketched strokes. The user tapped the undo button to return the deleted class and the sketched letter. However, the function returned another irrelevant class from a geometrical figure to a sketch. Refer to figure 15. Occurrence: 1.</p>
3. Overlap between sketches and geometrical objects.	<p>Issue 1: After transforming a sketch to a class the remaining sketch which was not recognized whether it was random strokes or text will be hidden behind classes. Refer to figure 12. Occurrence: 3.</p> <p>Issue 2: The sketched text hidden behind the classes overlaps once the classes are moved. Refer to figure 13. And in some cases it overlaps the edit windows. Refer to figure 26. Occurrence: 2.</p>
3. Converting feature needs improvement	<p>Issue 1: Currently when the user sketches a diagram, the user had to either convert every sketched class to a geometrical object directly after sketching it then add information to it. Or sketch the diagram without using the converting feature. That is because the converting feature is not at a level where it could understand a whole diagram (for example type or relationship, the text and where it belongs, etc.) Refer to figure A.14 in appendix a. Occurrence: 3.</p> <p>Issue 2: Sketching a class using more than one stroke can not be recognized. If a user uses multiple strokes to sketch a class then converting it to a geometrical figure will not work. That is because the feature processes one stroke at a time. Refer to figure A.19 in appendix a. Occurrence: 2.</p>

4. Results

4. Relationship arrows	The conversion feature does not recognize curved relationship arrows. Refer to figure A.16 in appendix a. Occurrence: 1.
5. Moving sketch	Moving sketched figures and text is not as consistent as moving geometrical figures. The users can sometimes move the sketch and sometimes they can not. Refer to figure A.25 in appendix a. Occurrence: 1.

General issues

1. Moving Screen Feature	The moving screen feature did not work as it should. The users tried to move the screen around but the feature did not respond. Refer to figure A.18 in appendix a. Occurrence: 1.
2. Placing a new class	Placing a new class overlaps other classes, and in this case it overlapped the Edit window. Refer to figure A.20 in appendix a. Occurrence: 1.
Saving the work	The tool currently takes a screenshot of what is displayed on the screen. The screenshot does not include all the data of a big diagram and the more zoomed out the screen the harder it is to read the text. Refer to figure A.29 in appendix a.

4.2.3 Suggested improvements

The various suggestions resulted from the researchers' observations which can improve the usability of the tool will be presented.

1. Image processing for recognition [Sketching]

To improve the sketching feature an overall change of the way it handles things needs to be made. Currently the tool processes each sketched stroke individually. This approach is limited e.g. the tool has managed to recognize strokes when the user tries to sketch a box or a relationship arrow using one stroke. However, sketching a box or a relationship arrow using more than one stroke cannot be recognized. An image processing approach can be more useful. The user will feel less restricted by the tool's restrictions and focus more on sketching. With image processing the tool will not only be able to recognize sketched figures with more than one stroke but it will also be able to recognize text. Recognizing text is important for such tools.

2. Introducing gestures and mimic the whiteboard experience [Geometrical figures, Sketching]

It was observed and discussed during the coffee breaks that gestures are important for such tools. Currently when using the sketching feature the tool does not feel intuitive to use. The only feature which seems to be intuitive to the user while using the sketching feature is to sketch. Moving the screen, zooming in and out,

and deleting functions are not intuitive to the user. Introducing gestures can make the tool more intuitive. For example introducing pinch to zoom gesture can feel more spontaneous to the user since it is being used widely on touch screen devices. Using the hand to move the screen could also feel spontaneous to the user. And also introducing hardware objects like an eraser to erase sketch and sketching with a pen can also feel more intuitive and mimics the whiteboard experience. If the user sketches with a pen he can move sketches with his finger which could be more user friendly.

3. Introducing on screen keyboard in a more usable way [Geometrical figures]

Currently the tool has an on screen keyboard. However, on a large screen where a keyboard takes half of the screen is not a very user friendly way to use an on screen keyboard. It should be made so that the keyboard only pops up when needed. For example when user clicks a textfield a small keyboard appears under the textfield. The on screen keyboard which is currently being used on windows 10 is resizeable. And it can also pop up when needed for example when the user taps on a textfield. It would also be nice if multiple users can bring up more than one keyboard at a time when they open multiple Edit windows on a large screen.

5

Discussion

This section will discuss findings of the study and discuss what factors influenced the findings. The section will also provide guidelines for creating an enhanced UML tools with smartboard as an environment. There will be two subsections one for each research question.

5.1 Discussing the Effect of Visual Appearances On the Design Process (RQ1)

This subsection will discuss the findings which supported the first research question. Factors which affected the findings will be discussed and how the usability evaluation findings correlate with the findings from the observational study.

5.1.1 Effectiveness and Efficiency

The result section presented a list of 4 possible implications, which will be discussed in this section.

1. *Geometrical figures feature is more effective than the sketching feature when dealing with an analysis task.*
2. *Overall, the geometrical figures feature is more effective than the sketching feature.*
3. *Sketching is more efficient than the geometrical figures for design tasks.*
4. *Geometrical figures is more efficient than the sketching feature when dealing with an analysis task.*

Even though all of the four implications differ in its specific scenario e.g. only analysis task or overall for each feature, they all had the same factors which affected them. As seen in the result section the geometrical figures feature was implied as being more effective than the sketching feature. 4 out of 8 comparisons implied using geometrical figures was more effective while there were 2 cases which implied sketching feature being more effective and there were 2 comparison where no conclusion could be drawn. This is a possible implication that geometrical figures feature is more effective since it has 4 out of 8 comparison. The comparisons and its measurements do imply the level of effectiveness but there are several factors which affected the effectiveness of each feature beside the feature itself.

The most common affecting factor is, the participants themselves. The level of the participants skill had a definitive impact on the score, but this validity was reduced by choosing participants who have at least passed an advanced class in

software architecture. Even though with the attempt of choosing participants who have approximately the same experience in architecture, the level of their skill itself could have impacted the task's solution.

The design flaws and improper functionality of each feature were a small impact on the measurements. Even though both feature had issues like moving objects, UI overlapping and different steps of adding information in classes. It should not be a big impact on the measurements because the features are sufficient enough to create proper class diagrams with the right information, association etc. However, the improperness of the features might have affected the participants by stressing them out during the development process or might have a negative impact on the participants themselves by demotivating them.

The number of participants of the study limited the result. The usability evaluation should be considered as possible implications because the number of data points did not allow the study to draw a concrete conclusion as to which feature is more effective and/or efficient.

There were no possible way to conclude further stuff with data received from the usability evaluation by it self. therefore, the data received from the usability evaluation will discussed with the observational study

5.1.2 Understanding the Usability findings From the Observational Study findings

It was hard to draw conclusions from the data points to make a definitive conclusion as to which feature had a better usability. In section 5.1.1 four possible implications have been drawn.

1. *Geometrical figures feature is more effective than the sketching feature when dealing with an analysis task.*
2. *Overall, the geometrical figures feature is more effective than the sketching feature.*
3. *Sketching is more efficient than the geometrical figures for design tasks.*
4. *Geometrical figures is more efficient than the sketching feature when dealing with an analysis task.*

By looking through the data obtained from the observational study the implications can be supported further.

The first implication suggested that geometrical figures feature is more effective when dealing with an analysis task. This can be seen by the effectiveness measurements and by the participants' input. Using geometrical figures allows the participants to directly create and move around classes which is hard to do in the current state of the sketching feature. Issue with moving was observed during the study session and reported by the participants. Using geometrical figures is more readable than using sketching feature. In the case when a model becomes big it is harder to read it when it is sketched, thus, allowing the geometrical figures to be more effective in the matter of accuracy and readability.

The second implication implied that at an overall level geometrical figures feature is more effective than the sketching feature. This could be due to geometrical figures being more familiar and less inconvenient than the sketching feature due to

the sketching feature's design flaws. As seen from the effectiveness measurements and from the users qualitative input, geometrical was preferred. Many participants believed geometrical figures feature to be more stable and easier to manipulate than the sketching feature.

The fourth implication suggested that the geometrical figures feature is more efficient when solving an analysis task. By looking at the qualitative input from the users and our observations it can be seen that the usability of the sketching feature was not very user friendly. Deleting a sketch could be irritating and time consuming. Therefore, when solving an analysis task where the task requires more discussing and brainstorming, the users will tend to delete and sketch more than when solving a design task. Deleting more on the sketching feature could be the reason why the sketching feature is less efficient. Furthermore, the solution for an analysis task could become more complex and messy than the design tasks. Thus, it would require the users to move around and relocate their solution more and those functions are better on the geometrical figures feature.

As for the third implication suggested that the sketching feature is more efficient when solving a design task. A design task is more straight forward than the analysis task and does not require as much discussing as the analysis task. The deleting, relocating and moving functions could have been used less in the design task therefore the sketching feature is more efficient.

5.1.3 Satisfaction and SUS

The satisfaction in general goes from medium to high, this can be observed by result of the question, *How satisfied are you with the tool in general?*, and the result of the SUS questions. Most questions had a majority which leaned towards positive satisfaction. Each question had at least middle to high positive score, but there were cases where some question leaned towards negative satisfaction. Beside these negative questions there were some negative data which was just a small minority even to the point of being outliers. However, this should not be disregarded, although the system got positive satisfaction levels, it still had negative scores. This could be directly linked with the issues of some functions. The improper functions had an obvious negative impact on the scoring, thus, making it important to fix the issues for higher satisfactory levels in the future. Furthermore, the specific questions which leaned towards negative satisfactory were:

- Q1, "I think that I would like to use this system frequently"
- Q5, "I found the various functions in this system were well integrated"
- Q8, "I found the system very cumbersome to use"

However, not all of the three questions were negative in each scenario i.e. the tool, sketching feature and geometrical figures feature. In the scenario of the tool, Figure 4.6 and 4.7, Q1 and Q5 leaned towards negative satisfaction. The result of Q1 and Q5 were affected by the flaws of the system and Q1 was further affected by the lack of UML notations which made the participants reluctant to use the tool frequently. In the scenario of the sketching feature, Figure 4.8 and 4.9, all of the above three questions Q1, Q5 and Q8 leaned towards negative satisfactory. This was due to the flaws and bugs of the sketching feature, which made the participants unsatisfied in

these three specific questions. In the scenario of geometrical figures feature, Figure 4.10 and 4.11, only Q1 leaned towards negative satisfactory, this was due to the lack of UML notation for geometrical figures feature, the users could not see themselves using such a tool frequently because the available notations are limited. From the SUS results it could be seen that the participants are more satisfied in using the geometrical figures feature since only 1 question had a negative satisfactory level compared to the sketching feature which had 3.

5.2 What the Users Would Like to See in a Tool Which Combine the Use of Two Different Visual Appearances (RQ).

To address the question, “*What features users would like to see when sketching or using geometrical figures to design a UML model?*”, observations and qualitative data have been gathered and presented in the results section. From the obtained results it can be seen that there are some design issues that were more severe than others. Thus, some design suggestions which can be used to create a functional and user friendly UML tool which utilizes smartboards will be presented and discussed.

5.2.1 Having a variety of notations [Geometrical figures, Sketching]

A usable tool should have many different UML notations. For the tasks used in this study, the users preferred having component diagrams, class diagrams and deployment diagrams. However, for other tasks more notations should be available to be used in the geometrical figures feature and recognizable when using the sketching feature.

5.2.2 Editing geometrical figures [Geometrical figures]

While using uml the users will most definitely need to add information to the geometrical figures. Having a small onscreen touch keyboard located around the editing windows which the user can access to add information to the geometrical figures seems like an intuitive design. When a user presses on a textfield the keyboard should appear, this is because the users are used to such a design when using touch screen devices e.g. when surfing on a smartphone.

5.2.3 Selecting objects [Geometrical, Sketching]

While it might not sound like an important thing to consider, selecting objects was one of the most frustrating design issues discovered in this study. Selecting an object whether it is a geometrical figure or a sketch should be designed in a smart way. The tool should allow the user to use the common method for selecting on touch screen devices. It requires the user to press and drag with one finger an area to select. And to add to the selection the user would just tap on other objects to be added.

If a user decides to deselect the selected classes the user would just have to press on an empty space to deselect. If a user started to use another function for example opening a class to add information or sketching with a pen the tool should deselect all the selected objects. This is because the user tends to forget to deselect and continue with his work and when the user wants to move his new object he would end up dragging other selected objects. The only two functions in which the tool should not deselect are when the user is moving around the screen and when the user is zooming in and out. This is because the user could be intentionally using those two functions to add more objects. Finally, when an object is selected it should be highlighted enough to indicate to the user that objects are selected. The color used for selecting should be consistent for both sketching feature and geometrical figures feature.

5.2.4 Having dynamic relationship arrows.

The relationship arrows are an important notation in UML, having an intuitive usability for it is a necessity. A few suggestions to improve the usability of the relationship arrows will be presented.

Editing relationship arrows. The user usually tends to make the relationship arrow without deciding on the type of the relationship and decide later on during his work. An intuitive solution would be to allow the user to tap on one end of the relationship arrow and add information. A small window should appear displaying the different type of relationships and multiplicities and a textfield for naming the relationship arrow which is shared between the two ends.

Readability of relationship arrows. The relationship arrows should never overlap each other when they are connected to a class. A common solution to such a problem would be to have more anchors to a class so that the relationship arrows would not overlap. As for when a relationship arrow crosses other relationship arrows belonging to other classes. There should be a simple indicator so the user would not mix between the two relationship arrows. The tool should locate the naming and multiplicities of the relationship arrows in a smart way where it would also not overlap other relationship arrows.

Relocating relationship arrows. On a different note, yet still related to the relationship arrows; the relationship arrows should not be static. If the user needed to connect a relationship arrow to another class, he should not need to delete the relationship arrow and create a new one. It should be allowed to drag a relationship arrow from one anchor and connect it to another. To make the usability of this feature more efficient the relationship arrows should become relocatable.

5.2.5 Gestures [Geometrical figures, Sketching]

It is necessary for a tool which is aiming to mimic the whiteboard experience, to have features which support that. The current existing tools seem like any other UML tool which uses touch instead of mouse and keyboard to place geometrical figures. Introducing gestures could be a huge improvement towards enhancing the user experience. A few suggestions as to which gestures are useful will be presented.

Erasing. To mimic the whiteboard experience and make the erasing feature intuitive to the user erasing a sketch using a hardware/physical tool is needed. Using buttons when sketching is not intuitive.

Moving the screen. To allow the user to move around a large screen, it needs to be made in a smart way. Since most of touch screen devices support multi touch, Moving the screen with 3-5 figures can be more intuitive to the user. And since the fingers are close to each other, adding this gesture should not collide with other gestures.

Pinch to zoom. Pinch to zoom is a good feature to add overall. Due to the fact that it is being used in all touch screen devices, the user expect the existence of such feature. The pinch to zoom gesture can work with two fingers or two hands. With the current suggestions, it should not collide with other gestures or functions.

Double tap/ long tap. Double tapping or long pressing on the screen to open a small menu is suggested to be added. The small menu could have copy and paste options to recreate a sketch or a geometrical figure. This could improve the efficiency because the user would copy and paste instead of spending time creating the same objects.

5.2.6 Undo and redo [General]

Undo and redo are basic functions which all the tools should have. There was no new intuitive way that could be introduced. Therefore, the suggestion would be to have an undo and redo buttons on the function bar which the user would use when needed. This is because most of the sketching/painting tools have the undo and redo functions on the function bar and the users are familiar with such design.

5.2.7 Saving the work [General]

When the user creates a design the tool should provide some methods to save the work. The tool should be able to produce a high resolution image of the work which covers the whole design or a selected part as the user requires. Also the tool should be able to store the current project for future editing.

5.2.8 Sketch recognition & Sketch transformation [Sketching]

The key feature for a tool which combines sketching and geometrical figures is a good sketch transformation. The sketch recognition feature needs to be able to recognize different type of shapes, different types of relationship arrows and text. An image processing combined with a text recognition algorithm could be used for analyzing the sketched diagram. The recognition can run in the background as well and highlights the recognized sketch for example green so the users would know that their sketch is recognized. Also when a class is sketched the tool should understand that this is a class and offers to move or convert the sketch of the class along with the contents of the class if needed. After the user finish his sketching he can convert the whole diagram and turn it into geometrical figures diagram.

5.2.9 A Sketch and geometrical figure view [Geometrical figures, Sketching]

In the case of a sketch overlapping a geometrical figure there should be a button located around the overlapped area which the user can press to switch between the two overlapped objects and decide whether he wants to move or delete objects to remove the overlapping.

6

Validity Threats

The validity threats measure how true and unbiased the results are [23]. Four validity threats have been detected. Within this section these validity threats will be presented and the solutions made to minimize them.

Construct validity threat addresses whether the participants understood the questions and the tasks required to solve [23]. To reduce this validity threat a pilot test has been made. The pilot testers went through all of the tasks and the questionnaire required for the observational study. All of the issues discovered were addressed before the observational sessions started.

Another validity threat which could affect the study is learning. In the specific case where participants might learn the system better in the first task which boost their performance in the second task. This threat was reduced by giving the participants a tutorial of OctoUML to familiarize them with the tool.

External validity threat addresses whether it is possible to generalize the findings of a study and whether the finding interesting to people with similar studies [23]. OctoUML was chosen as a representation for tools which combine the use of sketching and geometrical figures. Replicating the study using the current state of OctoUML should produce the same results. Replicating the study using other tools might introduce different results if the environment was not controlled. E.g. OctoUML lacked in sketch recognition where another tool excelled. Replicating the study on that tool could produce higher effectiveness rates for the sketching feature if not controlled to match OctoUML.

Reliability validity threat addresses whether the data and analysis is dependent on the researcher [23]. Two researchers worked together to reduce the threat to the reliability. Each study session were conducted by both researchers to reduce any misunderstanding and bias from both parties i.e. researchers and participants. Furthermore, the researchers analyzed the recordings individually and compared the findings among themselves.

7

Conclusions & Future Work

The purpose of the study focuses on understanding the impact which different visual appearances, sketching feature and geometrical figures feature, have on the design process. And to figure out what features are needed to make the usability of tools which combine visual appearances more intuitive for the users. An observational study and a usability evaluation were conducted where 16 participants were asked to collaborate in pairs to solve design and analysis tasks using the sketching feature and geometrical figures feature. This was done to have an understanding regarding whether the type of the task would affect the design process when using the sketching feature and the geometrical figures feature. Also to find out which feature yield better results when solving an analysis task and why. From the results the three usability characteristics were measured and it was deduced that the geometrical figures feature surpassed the sketching feature in terms of effectiveness and satisfaction. But in terms of efficiency, the sketching feature was more efficient than the geometrical figures feature. The geometrical figures feature was more effective because the readability of the diagram was less confusing and also the users were more used to using tools which utilize the geometrical figures feature. As for the satisfaction the users preferred the geometrical figures feature over the sketching feature because the sketching feature's functions e.g. deleting or moving a sketch were not very user friendly. The sketching feature was more efficient due to it being more flexible. The feature does not require several steps to create and add information in a class compared to the geometrical figures feature. In the discussion section a list of suggested guidelines deduced from the users' suggestions and the researchers' observation were presented. These guidelines can be used when creating tools which aim to combine the usability of the sketching feature and the geometrical figures feature.

More observational studies need to be conducted in the future to evaluate the usability of such tools. The conducted study mainly focused on using class diagram notations. The usability of different notations should be studied to discover new design issues and future improvements. Furthermore, another observational study needs to be made to validate the suggested guidelines and discover weaknesses and suggest new guidelines if required.

Bibliography

- [1] ISO/IEC 25010:2011, *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*, 2011.
- [2] Moody, D. (2009). *The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering*. IEEE Transactions on Software Engineering, 35(6), 756-779
- [3] Khaled, L. (2009, December). *A comparison between UML tools*. In *Environmental and Computer Science, 2009. ICECS'09*. Second International Conference on (pp. 111-114). IEEE.
- [4] Heena, & Garg, R. (2011). *A comparative study of UML tools*. In ACAI (pp. 1-4).
- [5] Di Rocco, J., Di Ruscio, D., Iovino, L., & Pierantonio, A. (2015). *Collaborative repositories in model-driven engineering [software technology]*. IEEE Software, 32(3), 28-34.
- [6] Seffah, A., Desmarais, M. C., & Metzker, E. (2005). *HCI, usability and software engineering integration: present and future*. In *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle* (pp. 37-57).
- [7] Lahtinen, S., & Peltonen, J. (2003, October). *Enhancing usability of UML CASE-tools with speech recognition*. In *Human Centric Computing Languages and Environments, 2003. Proceedings. 2003 IEEE Symposium on* (pp. 227-235). IEEE.
- [8] Blankenhorn, K., & Jeckle, M. (2004, September). *A UML profile for GUI layout*. In *Net. ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World* (pp. 110-121). Springer Berlin Heidelberg.
- [9] Chen, Q., Grundy, J., & Hosking, J. (2008). *SUMLOW: early design-stage sketching of UML diagrams on an E-whiteboard*. *Softw., Pract. Exper.*, 38(9), 961-994.
- [10] Mangano, N., Baker, A., Dempsey, M., Navarro, E., & van der Hoek, A. (2010, September). *Software design sketching with calico*. In *Proceedings of the IEEE/ACM international conference on Automated software engineering* (pp. 23-32). ACM.
- [11] Wüest, D., Seyff, N., & Glinz, M. (2012, October). *Flexisketch: A mobile sketching tool for software modeling*. In *International Conference on Mobile Computing, Applications, and Services* (pp. 225-244). Springer Berlin Heidelberg.

- [12] Damm, C. H., Hansen, K. M., & Thomsen, M. (2000, April). *Tool support for cooperative object-oriented design: gesture based modelling on an electronic whiteboard*. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (pp. 518-525). ACM.
- [13] Chen, Q., Grundy, J., & Hosking, J. (2003, October). *An e-whiteboard application to support early design-stage sketching of UML diagrams*. In Human Centric Computing Languages and Environments, 2003. Proceedings. 2003 IEEE Symposium on (pp. 219-226). IEEE.
- [14] Iivari, J. (1996). *Why are CASE tools not used?*. Communications of the ACM, 39(10), 94-103.
- [15] Nielsen, J. (2003). *Usability 101: Introduction to usability*.
- [16] International Organization for Standardization. (1998). *ISO 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs): Part 11: Guidance on Usability*.
- [17] Net, U. (2006). *International standards for HCI and usability*. Retrieved March, 22, 2006.
- [18] Bobkowska, A. E., & Reszke, K. (2005). *Usability of UML Modeling Tools*.
- [19] Green, T. R. G., & Petre, M. (1996). *Usability analysis of visual programming environments: a 'cognitive dimensions' framework*. Journal of Visual Languages & Computing, 7(2), 131-174.
- [20] Jolak, R., Vesin, B., Isaksson, M., & Chaudron, M. R. (2016). *Towards a new generation of software design environments: Supporting the use of informal and formal notations with octouml*. In Second International Workshop on Human Factors in Modeling (HuFaMo 2016). CEUR-WS (pp. 3-10).
- [21] Saldaña, J. (2015). *The coding manual for qualitative researchers*. Sage.
- [22] Birks, M., Chapman, Y., & Francis, K. (2008). *Memoing in qualitative research: Probing data and processes*. Journal of Research in Nursing, 13(1), 68-75.
- [23] Runeson, P., & Höst, M. (2009). *Guidelines for conducting and reporting case study research in software engineering*. Empirical software engineering, 14(2), 131.
- [24] Privitera, G. J. (2013). *Research methods for the behavioral sciences*. Sage Publications.
- [25] Brooke, J. (1996). *SUS-A quick and dirty usability scale*. Usability evaluation in industry, 189(194), 4-7.

A

Appendix 1

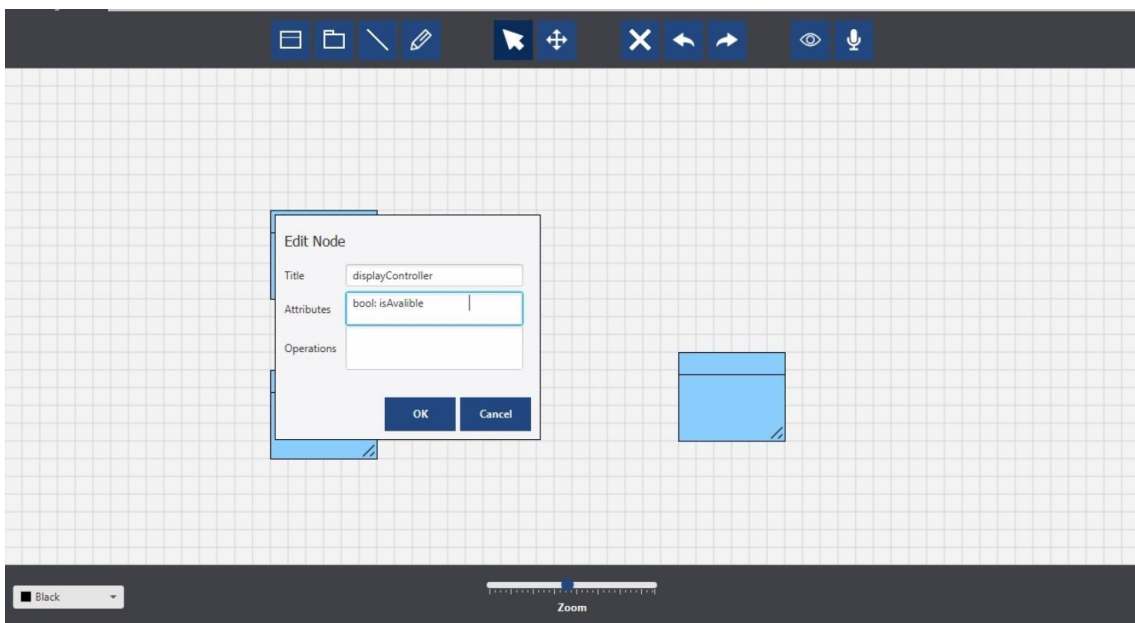


Figure A.1: Edit window issue, User can tab from title to attributes but not from attributes to operators.

A. Appendix 1

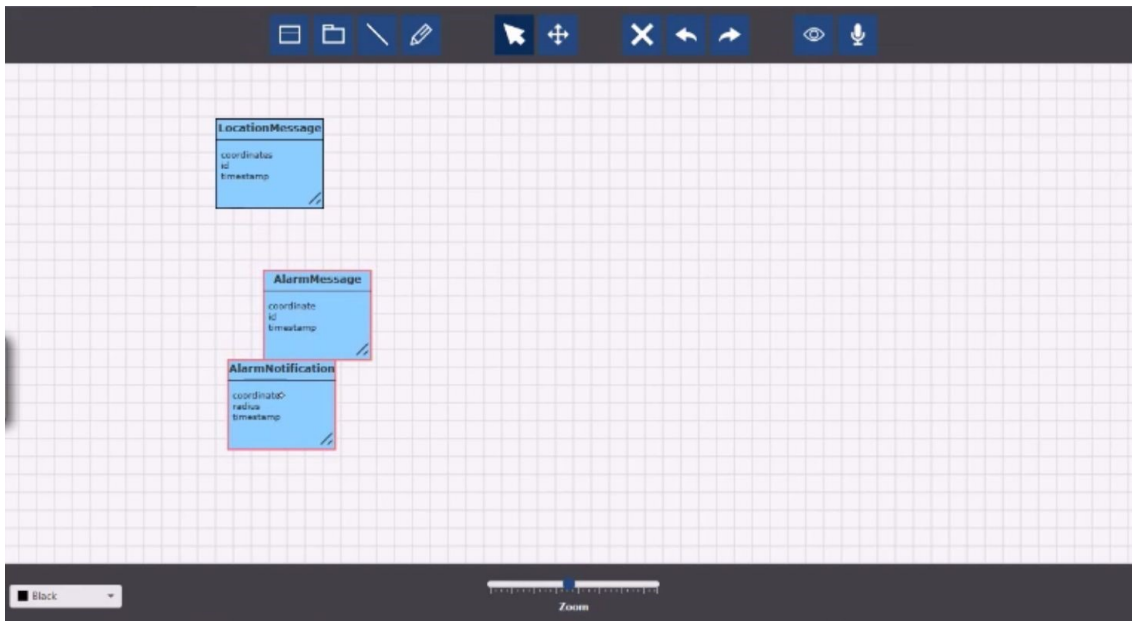


Figure A.2: The users dragged many classes because they forgot to untoggle the selected classes.

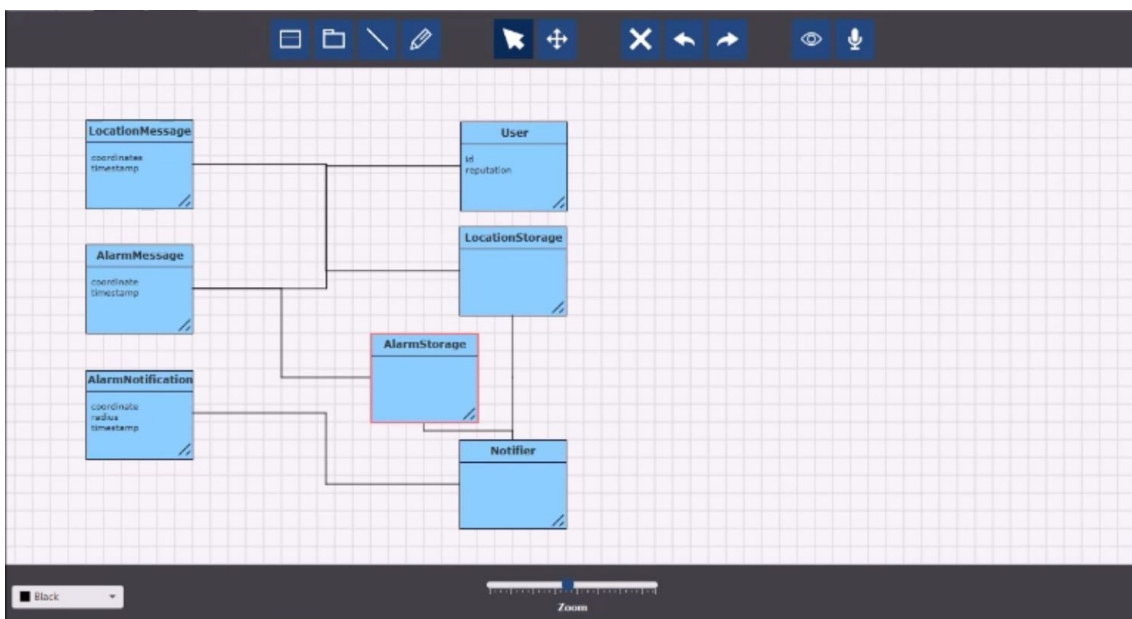


Figure A.3: Relationships arrows overlapping issue.

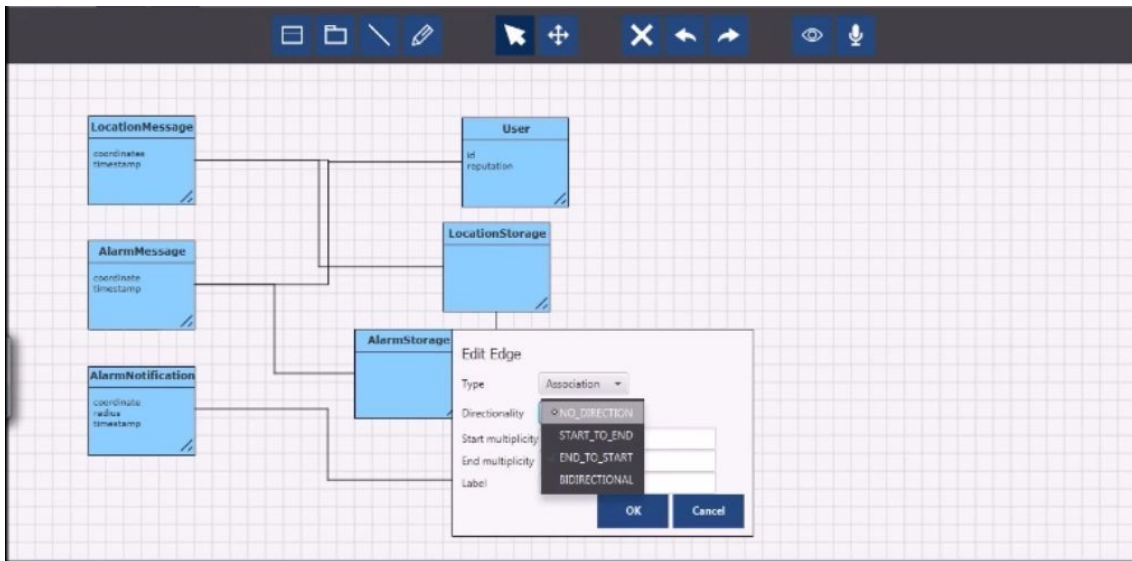


Figure A.4: Users are having issues remembering/Choosing the right direction for relationship arrows, the same goes for the multiplicities.

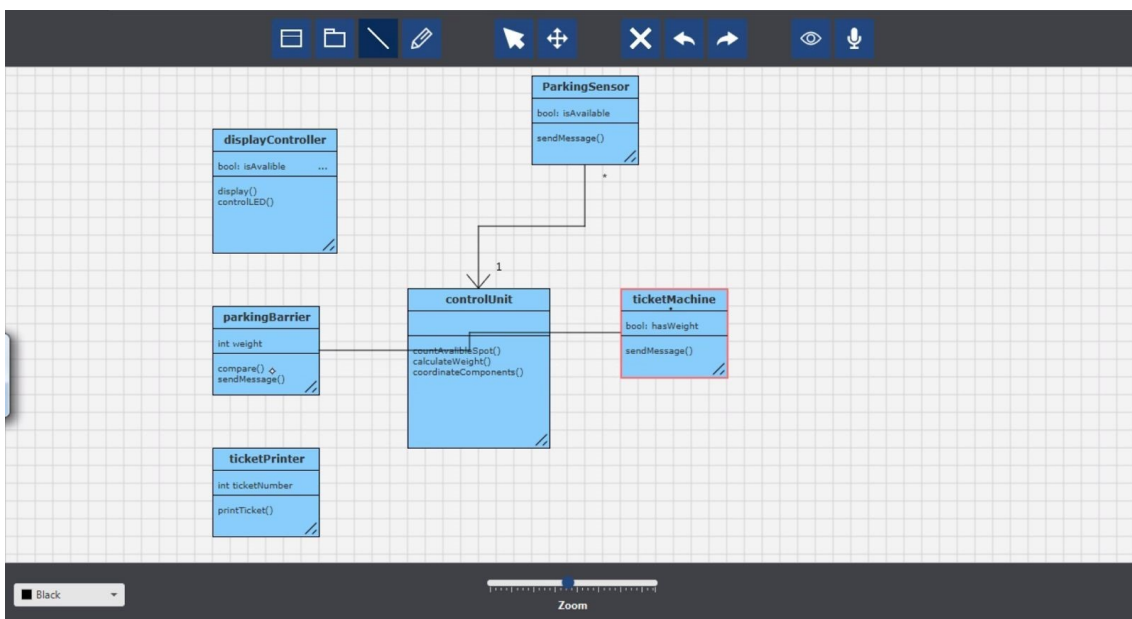


Figure A.5: The relationship arrow is overlapping a class.

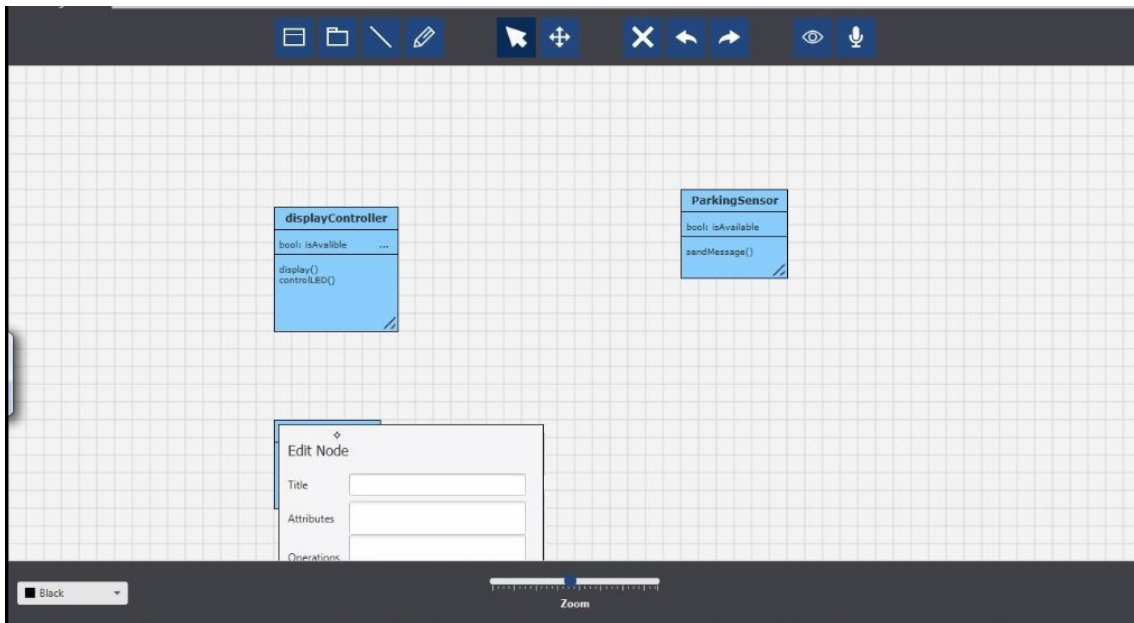


Figure A.6: The Edit window is cropped.

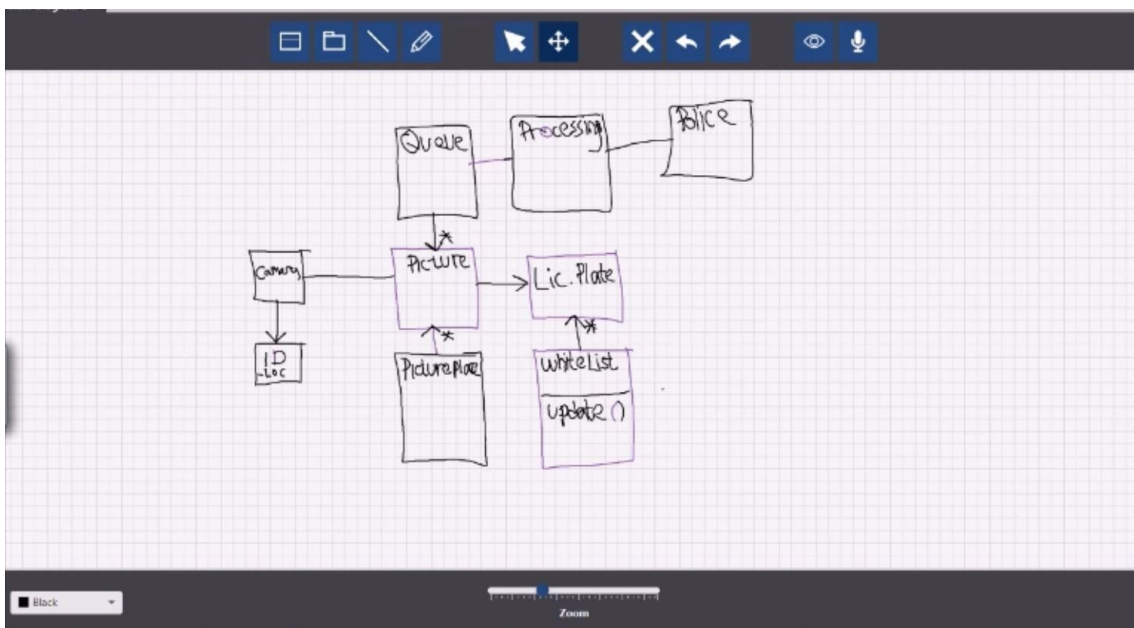


Figure A.7: The toggling issue in the sketching feature, the tool selects strokes by accident.

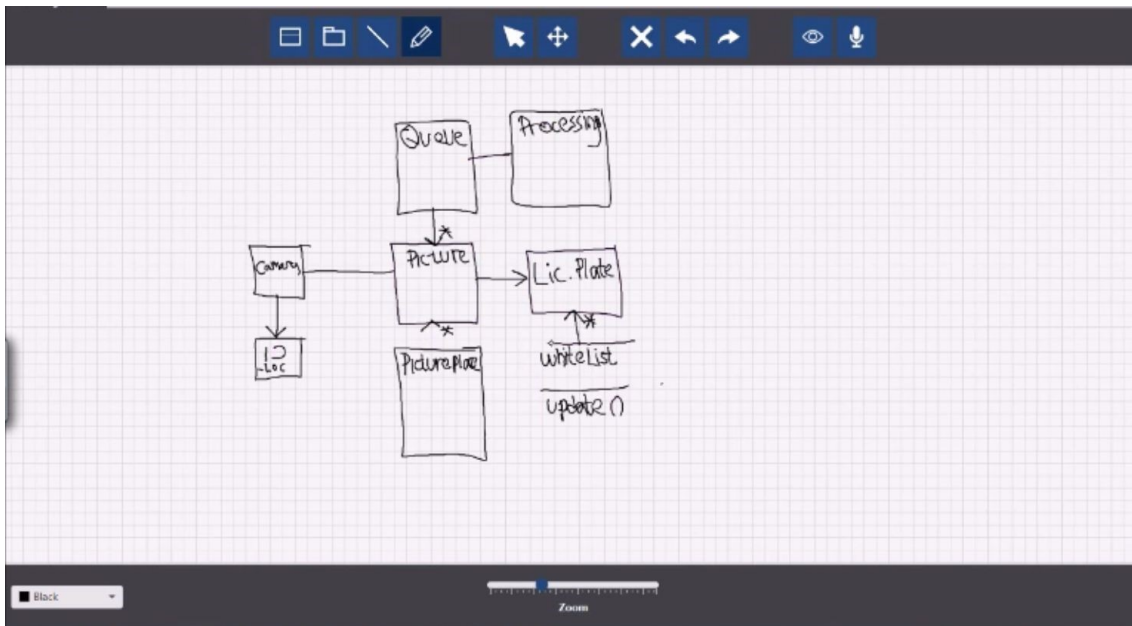


Figure A.8: Delete issue, the tool deleted strokes which were not selected.

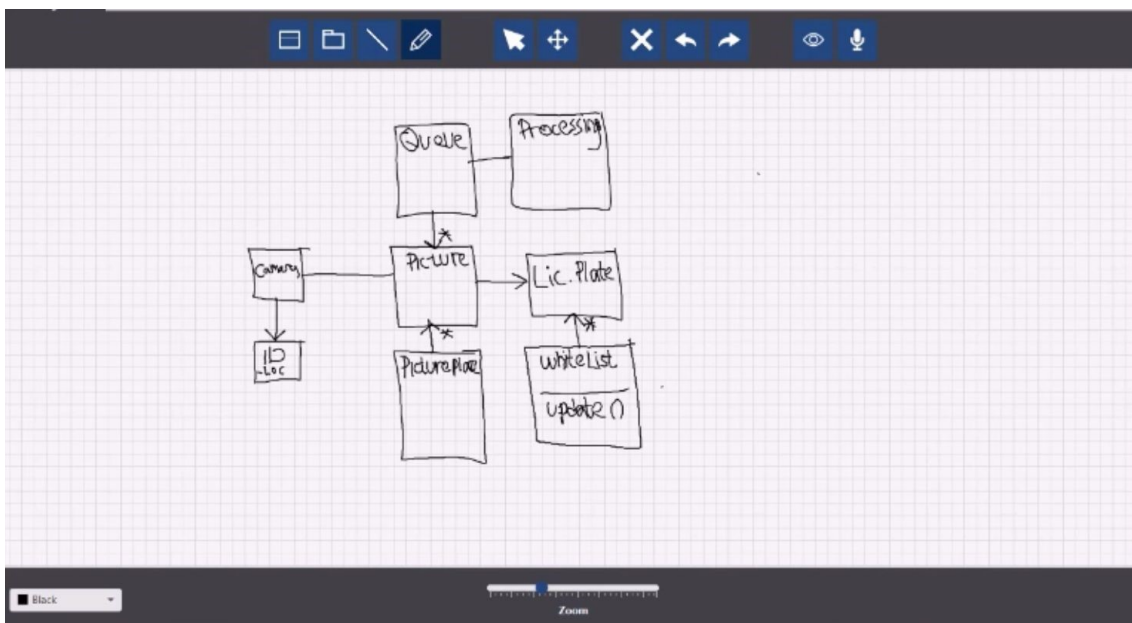


Figure A.9: Undo function did not work properly.

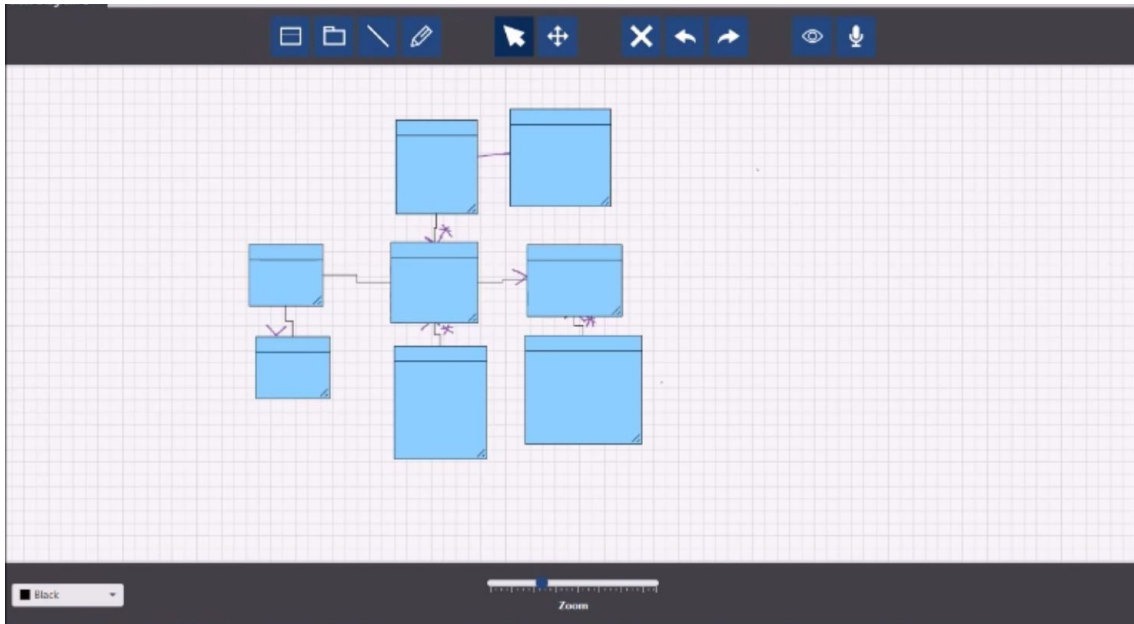


Figure A.10: Classes overlap sketched text.

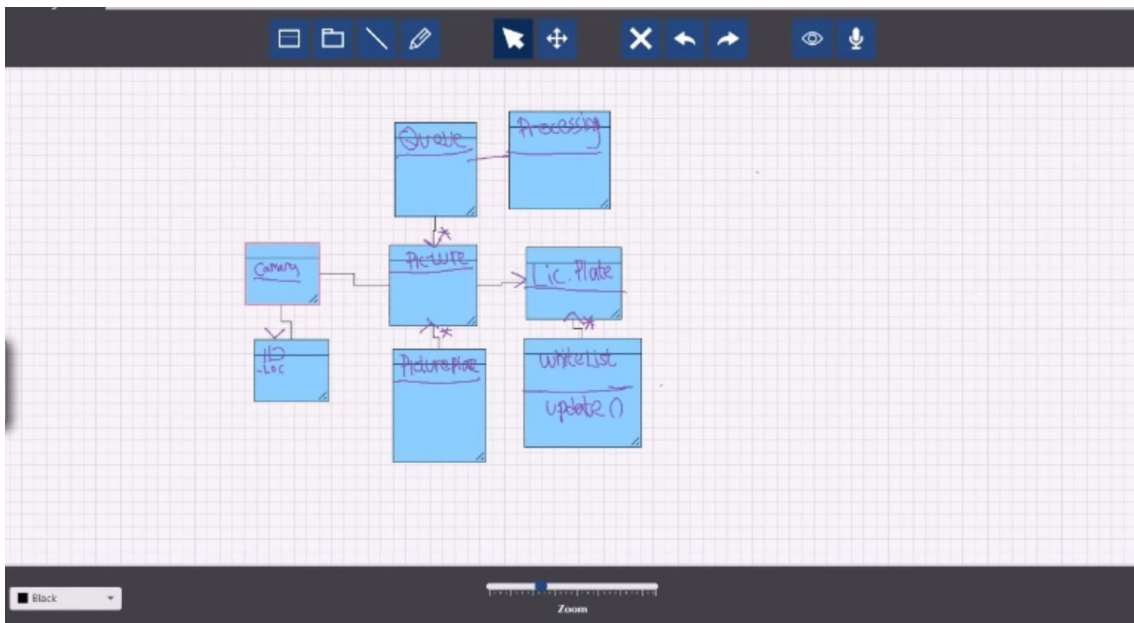


Figure A.11: Sketched text overlaps classes after dragging it.

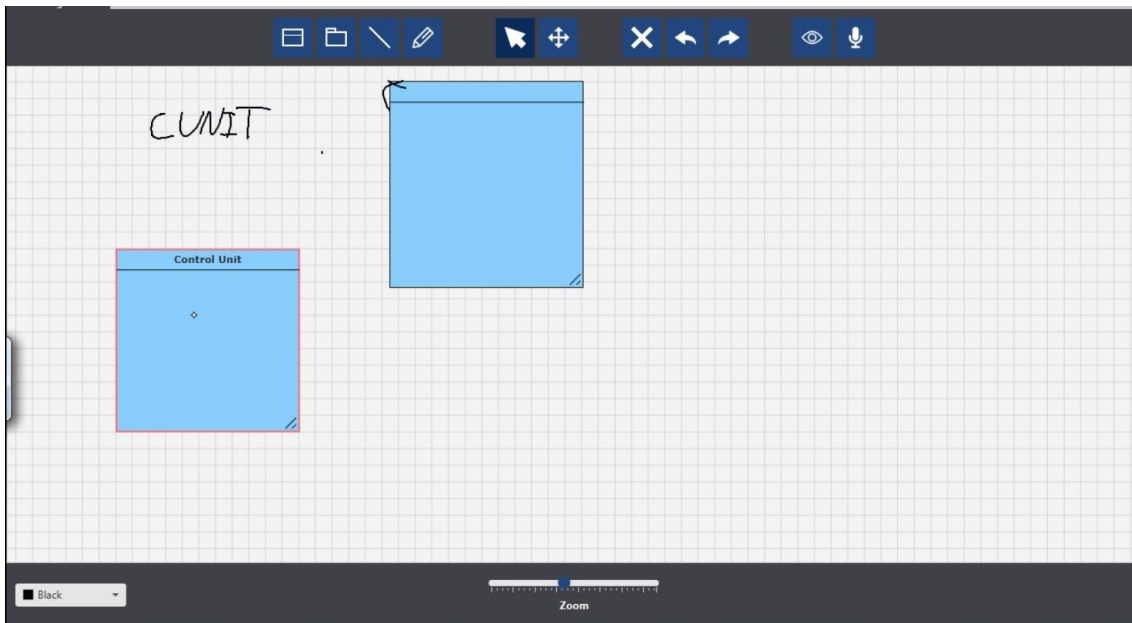


Figure A.12: The users had to convert the sketched class to a geometrical object directly after sketching it.

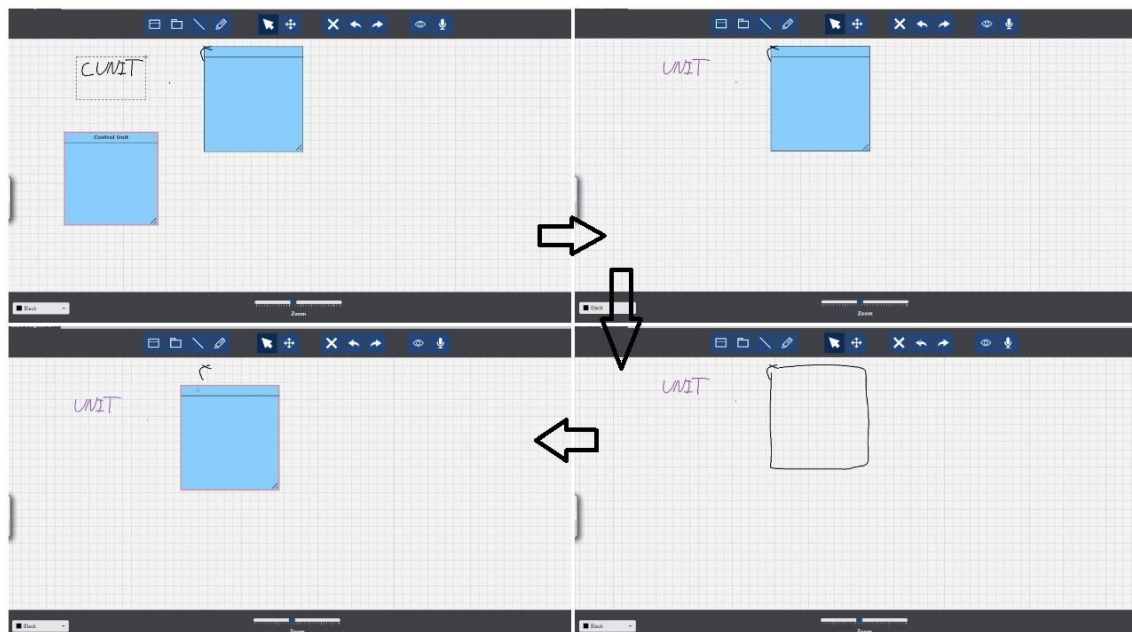


Figure A.13: The users tried to delete the sketched text and it deleted a class, then they tried to undo to return the deleted class and it turned another class to sketch. Then they used the redo button and it returned the second sketched class to a geometrical class.

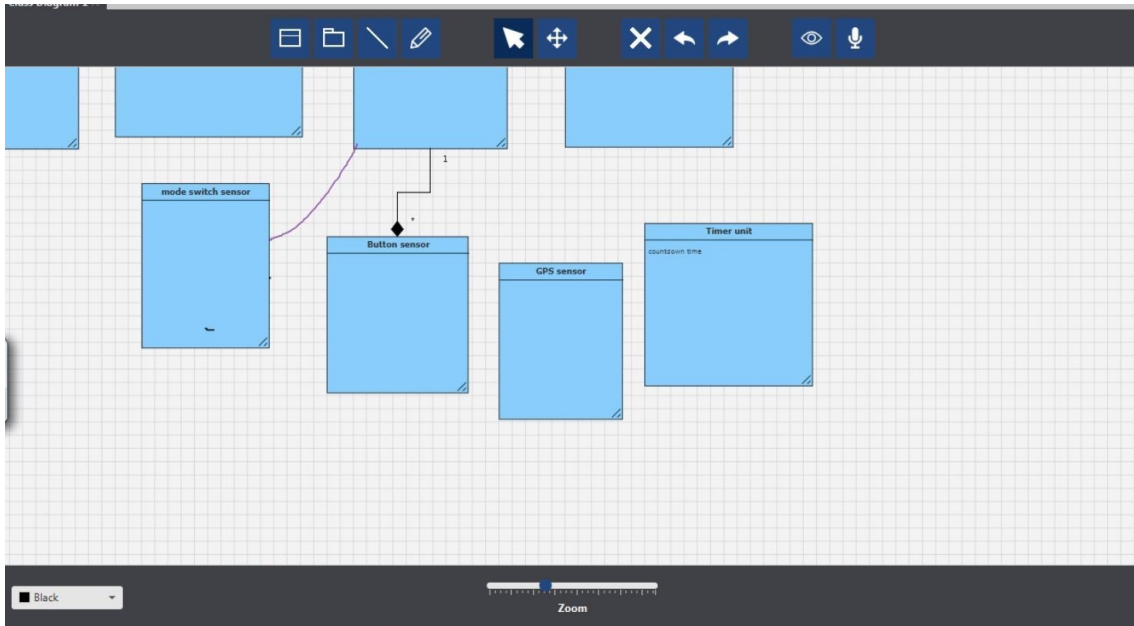


Figure A.14: Curved relationship arrows are not recognized.

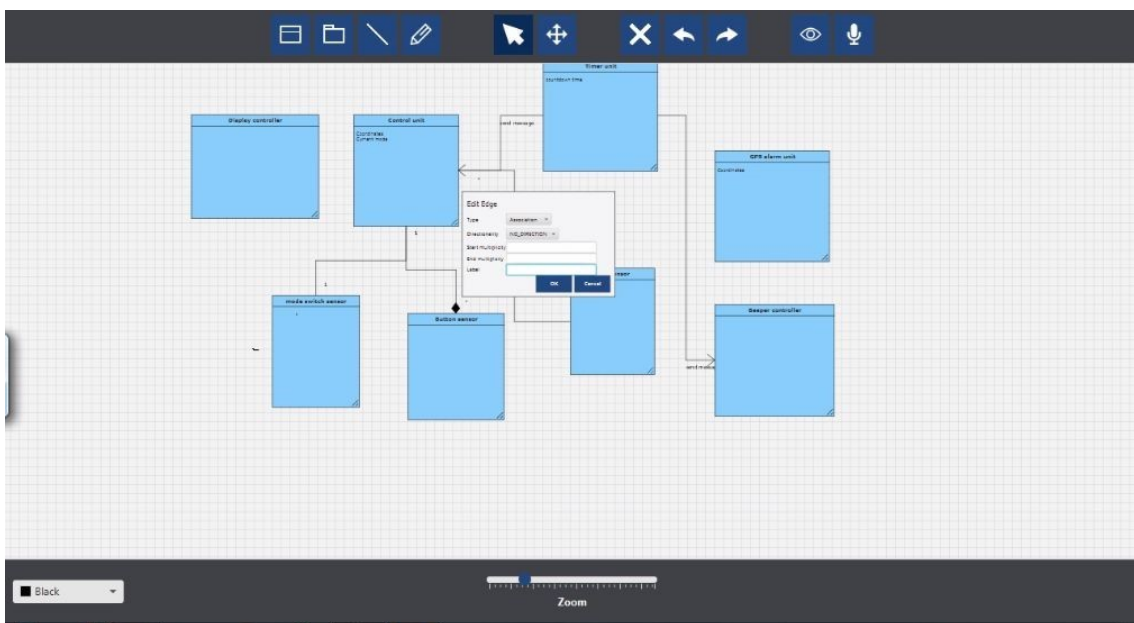


Figure A.15: The users would like to close the edit windows by pressing the Enter button on the keyboard.

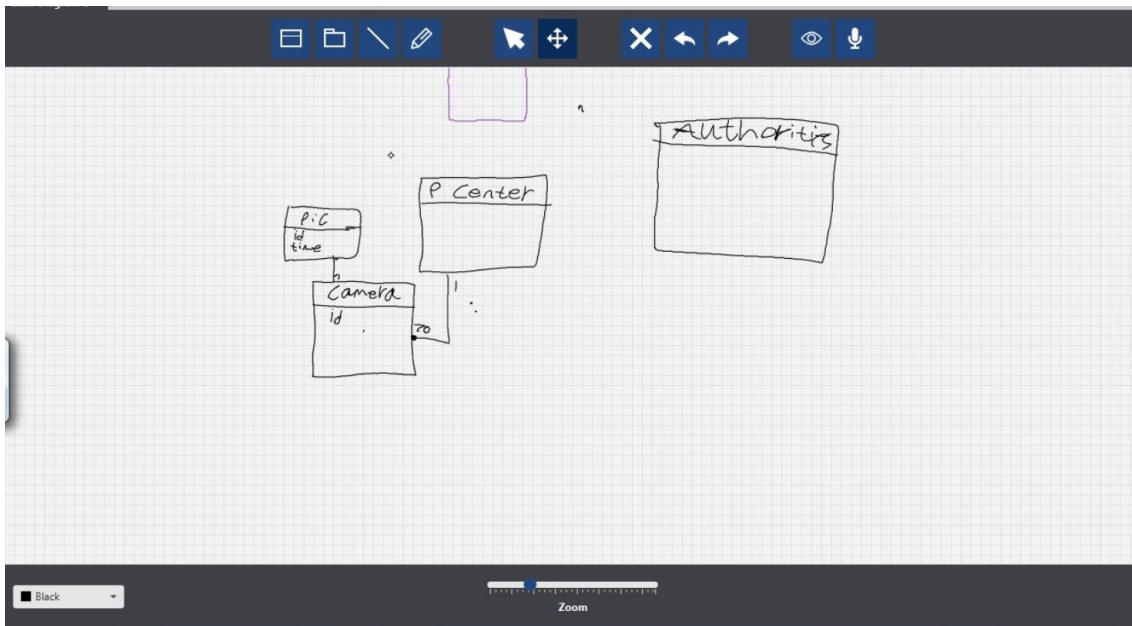


Figure A.16: The moving screen feature did not function properly. The users had to zoom out to have more space.

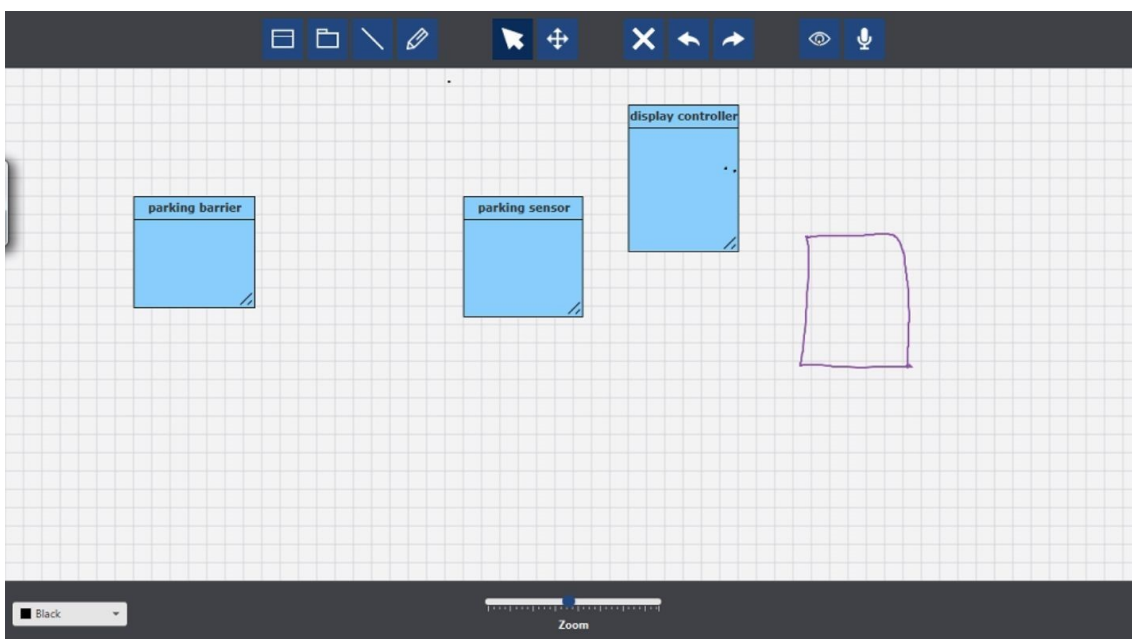


Figure A.17: The converting features does not recognize a class if it was made using more than one stroke.

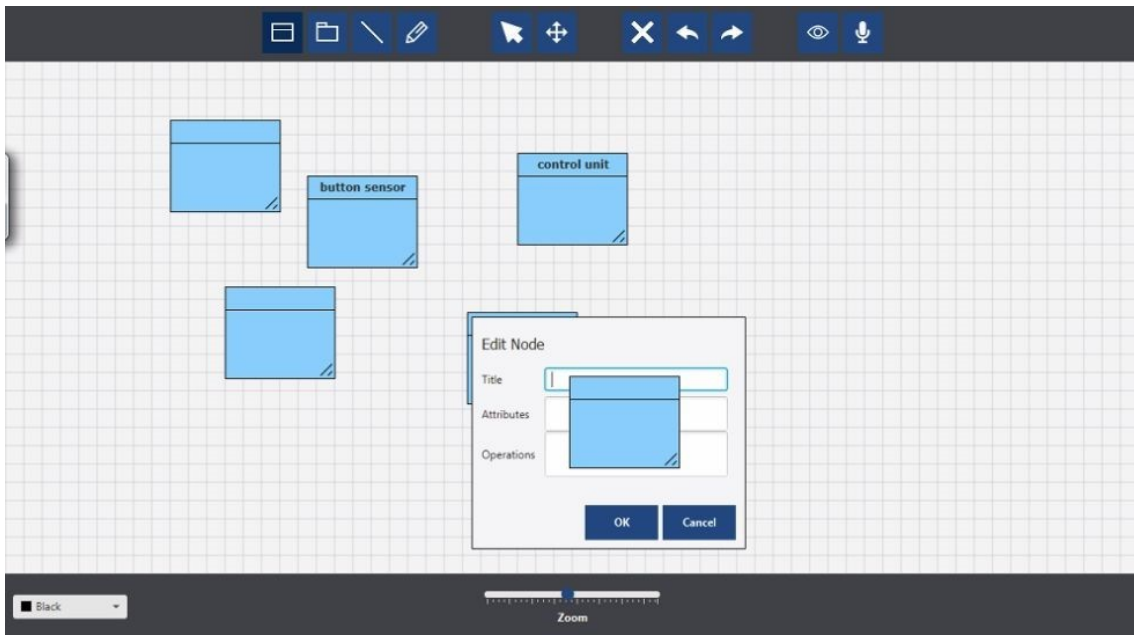


Figure A.18: A class was placed into the screen but it overlapped the edit window.

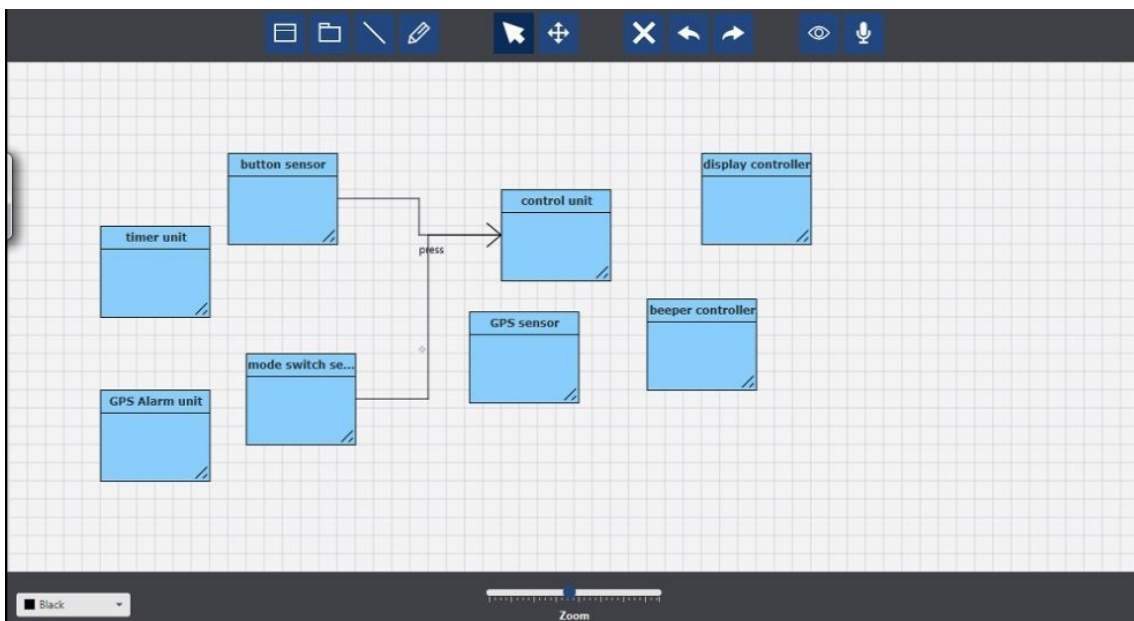


Figure A.19: A relationship arrow can overlap the relationship name of another relationship arrow.

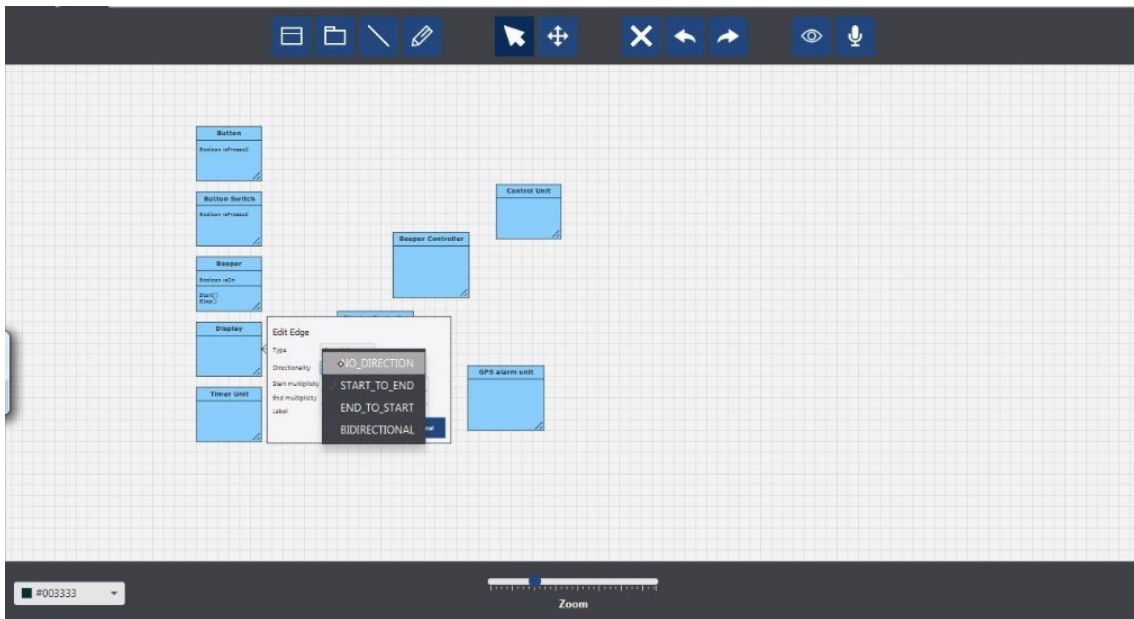


Figure A.20: The user tried to select the direction for the relationship arrow but it took three attempts for the tool to register.

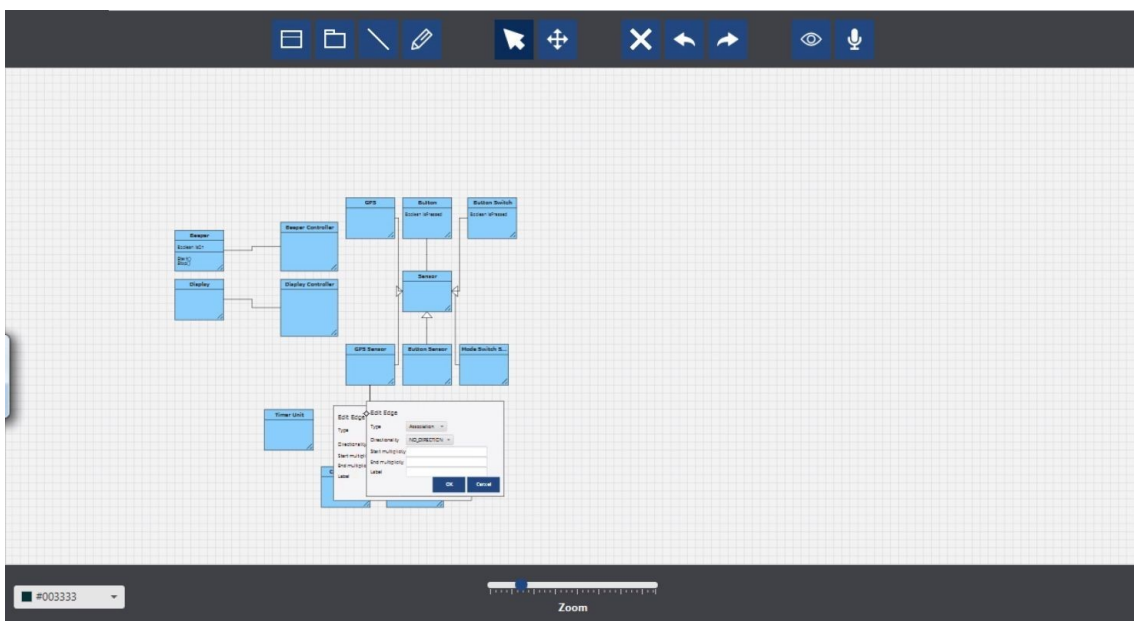


Figure A.21: Double tapping on combined relationship arrows will open two Edit window.

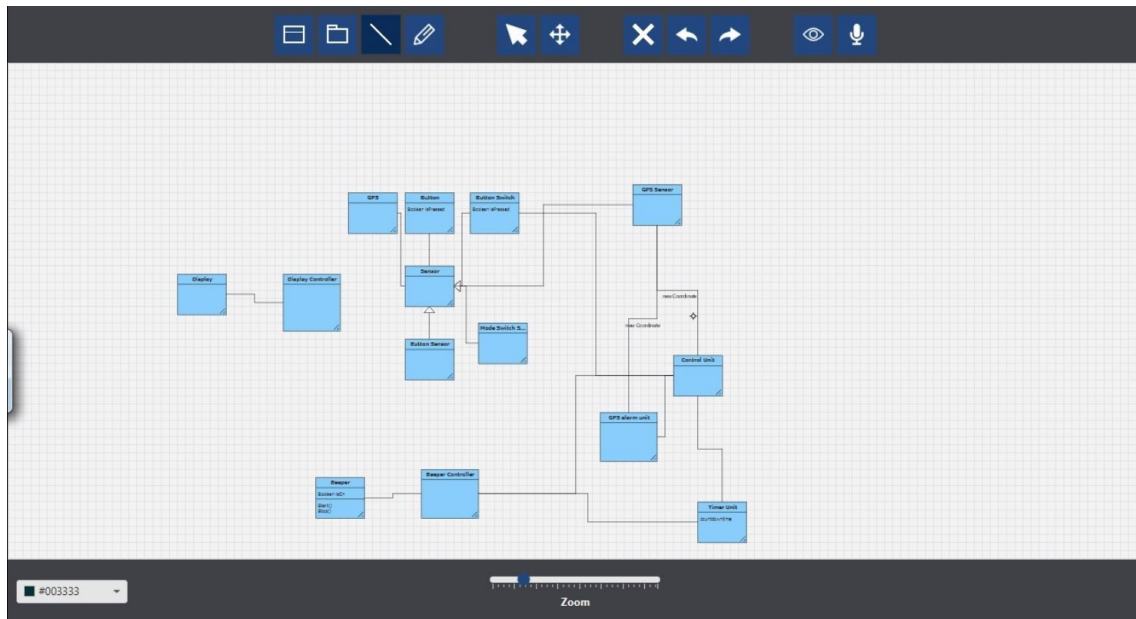


Figure A.22: The user can not select a relationship arrow to remove it.

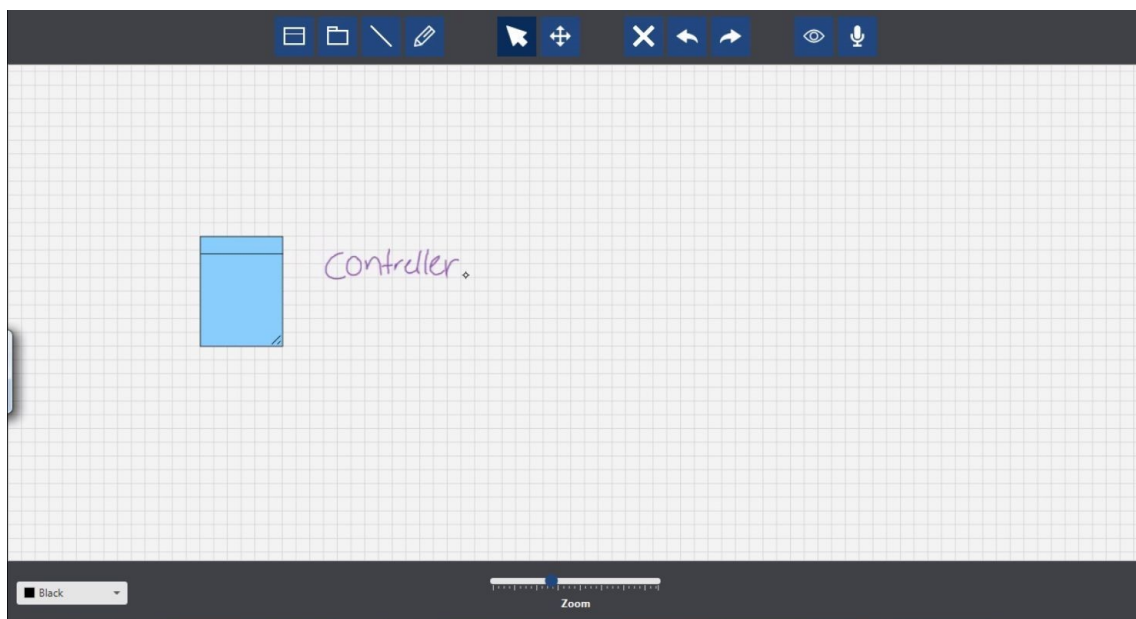


Figure A.23: The user tried to move a sketched text around but the moving is not as consistent as moving geometrical objects.

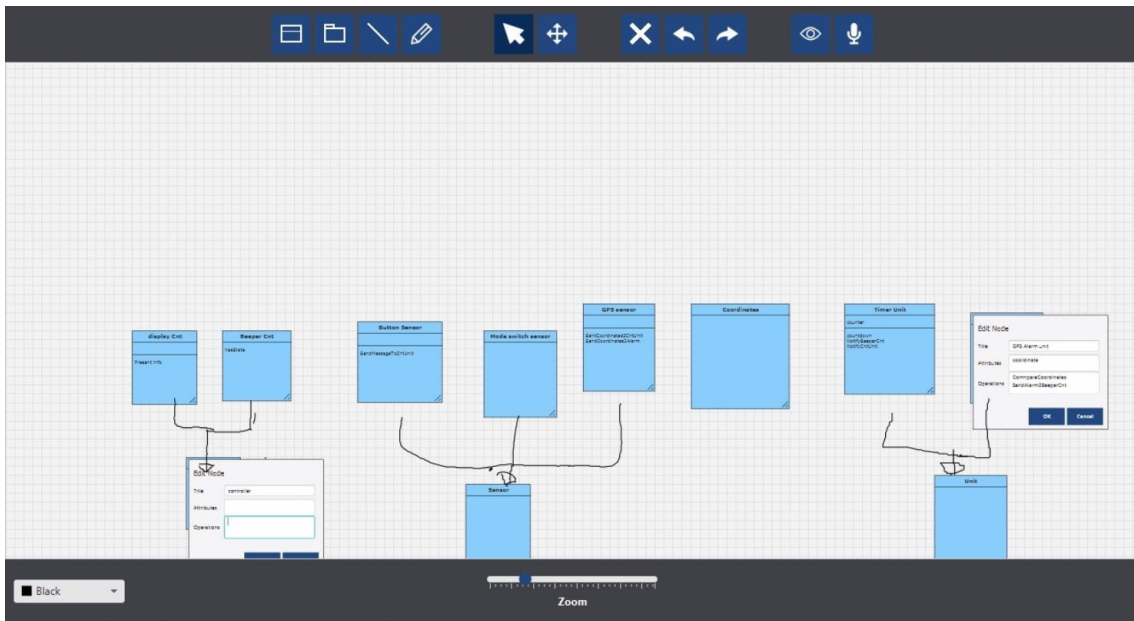


Figure A.24: The sketched strokes overlap classes and edit windows.

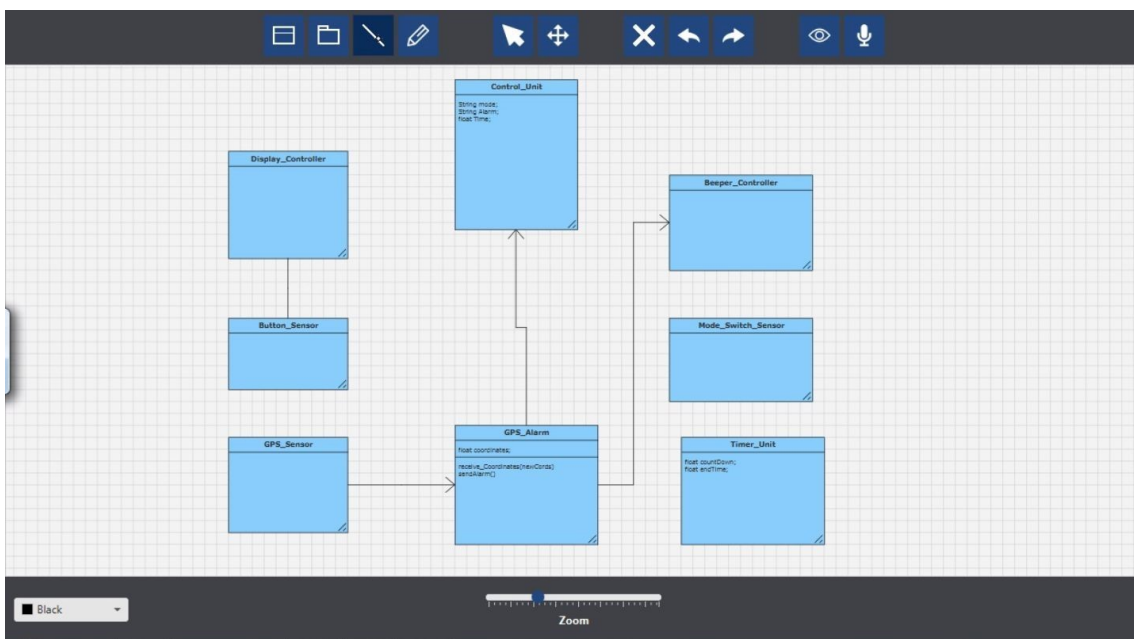


Figure A.25: The users did not like the current design used to remove the relationship arrows.

A. Appendix 1

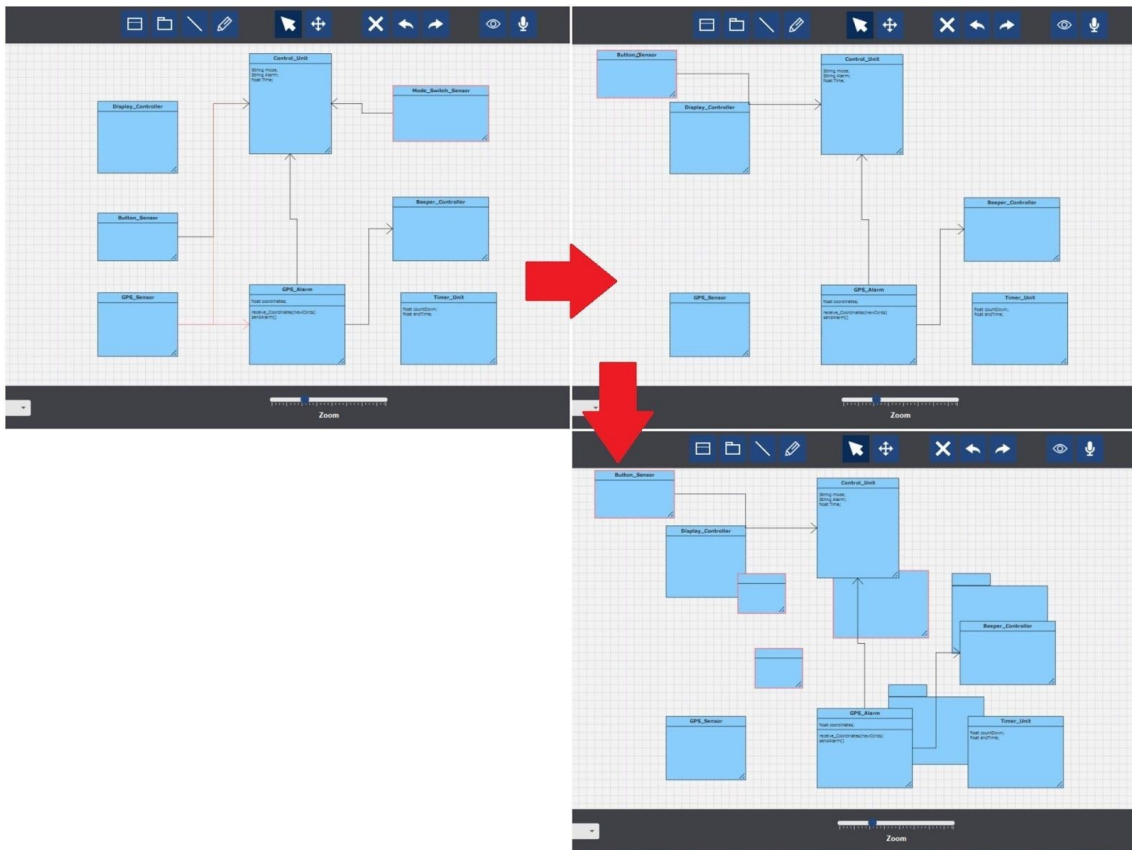


Figure A.26: The user tried to use the undo function but it did not work as it should.

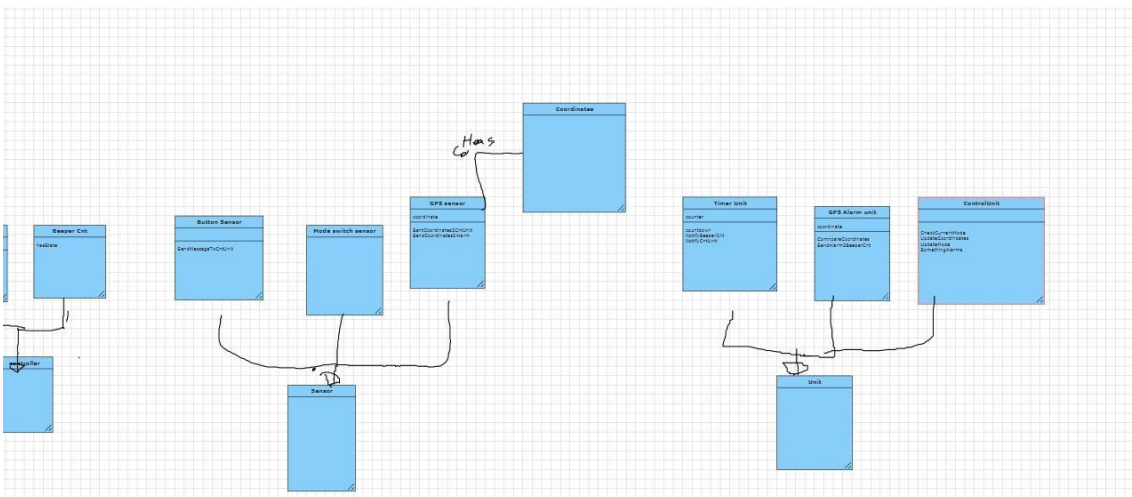


Figure A.27: A saved screenshot of a diagram obtained from the tool, some of the classes are cut and it is hard to read the information.

B

Appendix 2

B.1 Observational Study Session Tasks

B.1.1 Design task 1:

Design task 1: The city center of Gothenburg is crowded for most of the time. The municipality decided to create an app which assists the users to locate empty parking spaces. The parking app has the following software components. A display controller responsible for showing information on the display. A beeper controller that beeps repeatedly when started, and that is silent when it is stopped. A button sensor for each button that sends messages to a control unit whenever a physical/-touch button is pressed. A mode switch sensor that sends a message to the control unit whenever the switch changes mode. A GPS sensor that sends coordinates to the control unit and a GPS alarm unit. Whenever the coordinate changes it will be sent to the control unit and the GPS alarm unit as a parameter with the new coordinate. A timer unit that keeps track of and counts down the countdown time, and that sends a message to the beeper controller and the control unit when the countdown time reaches 00:00 00. A GPS alarm unit that maintains the set coordinates; it compares incoming actual coordinates messages with the set coordinate, and sends an alarm to the beeper controller and the control unit when the set coordinate is reached. The control unit keeps track of the current mode, and coordinates all other components in response to buttons pressed, mode switch changes, and alarms.

Create a representation of the system using Software Engineering diagrams.

B.1.2 Design task 2:

With municipality newly introduced app, Nordstan decided to introduce a new system for their parking garage, which manages parking. The system has the following software components. A display controller responsible for showing information (available/occupant etc) on the display and enabling/disabling the LED's of the parking signs. A parking sensor for each parking slot that sends messages to a control unit whenever a parking is occupied/available. A parking barrier: this is a scale which calculates the weight of the car and compare it to the threshold weight. Whenever an input weight occurs it sends a message to the appropriate controls for opening the barrier and when the weight returns to default it closes the barrier (this message has the calculated weight and the default weight as parameters). For

the “physical” ticket machine there is a corresponding “software” ticket machine that sends messages (prepare for incoming weight) to the parking barrier. A ticket printer can produce tickets at different entries and exits. The control unit keeps track of the amount of cars inside the park, and coordinates all other components and weight changes.

Create a representation of the system using Software Engineering diagrams.

B.1.3 Analysis task 1:

A city has about 1,000,000 inhabitants. The municipality wants to restrict the access of motorized vehicles to the inner city center. Therefore, the municipality has installed 70 cameras at the entry points to the city center. This perimeter is called the “red zone”. Only vehicles with a special permit are allowed to enter the red zone. Residents of the city center, city busses and qualified delivery trucks (e.g., to stock the shops) are allowed to enter the red zone. Also the police ambulances, fire trucks (and so on) are allowed. Other vehicles are in violation of the traffic law and need to be fined accordingly. The cameras take a picture of the license plate of the vehicles crossing the perimeter. Each picture also contains a timestamp and the identifier of the camera that took it. The identifier is associated to the location of the camera. These pictures are sent to a processing center. When received, each picture is analyzed via an optical character recognition (OCR) software and the license plates in each picture are extracted. The plate numbers need to be checked against the list of authorized vehicles. The system must provide a way to update the list of authorized vehicles. If an unauthorized vehicle is observed by the camera, the police are notified and the picture (and the contextual information about time and location) is also sent to them. The pictures and the information about the offending vehicles need to be stored in the system for legal reasons (e.g., in case of litigation) and it is important that such pictures are not lost for at least 2 years. During peak hours (e.g. at 5PM), it is expected that there will be up to 50 vehicles per second entering the red zone. The processing of a picture by the OCR software can take up to 2-3 seconds in order to achieve a good accuracy. The system should not let any unauthorized vehicle go unnoticed. However, it is not necessary for the system to inform the police in real-time about the violations.

Create a representation of the system using Software Engineering diagrams.

B.2 Analysis task 2:

DriveSafe is a startup developing a collaborative system to increase the safety on the roads. This is a pilot system initially developed for the city of Gothenburg but with the ambition to deploy it throughout Sweden. Functionality: The system consists of an app that car drivers can download on their smartphone and use to register to the service for free. For safety reasons, it is advised to keep the smartphone in a dashboard cradle. At regular intervals (e.g., every 30 seconds), the app generates a

location update message towards the DriveSafe Services. Such an update contains the current GPS location and the identity of the driver. The app offers a simple, non-destructive user interface with a big red button that can be pressed in case hazardous conditions are encountered by the driver on the road. When the button is pressed, the app generates an alarm message that is sent to the DriveSafe Services. The alarm message contains the current GPS location of the smartphone and the identity of the driver. At the DriveSafe Services, the alarms are stored and once several alarms from the same location are received (e.g., 2 in a short time interval from two different drivers) a warning message is sent out to all the vehicles in the same area (e.g., within a 1 km radius). The warning is displayed by the receiving app until the car leaves the area. The DriveSafe Services keeps a copy of the warning messages that are sent out. At the beginning, all drivers are considered as “trusted”. However, when a driver generates an alarm that is not confirmed by other drivers, their reputation is decreased. Once the reputation goes below a certain threshold (e.g. after two false alarms), the driver is labeled as “untrusted”. Alarms from untrusted drivers are discarded by the DriveSafe Services upon reception.

Create a representation of the system using Software Engineering diagrams.

B.3 Qualitative Questions

- When dealing with an analysis task what feature (Sketching or Geometrical) would you prefer to use?(Skip if you did not do analysis tasks)
- Did you have any problems completing the tasks? Elaborate, time, difficulty of task or difficulty of using the tools?
- Did you like the notations of OctoUML?Is there anything you would like to change (add/remove) on the notation of the feature(s)?
- What did you feel (negative/positive comments) about having the smart board as an environment?
- Which feature (Sketching or Geometrical) would you say is better? And why do you think it is better?

B.4 Quantitative Questions.

B.4.1 User satisfaction Question

On a scale of 1-5 how satisfied are you with the tool in general? (1 is lowest, 5 is highest)

B.4.2 System Usability Scale

1 is the lowest which indicates the users disagree and 5 is the highest which indicates the users agree.

- I think that I would like to use this system frequently.

- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.