ASSESSING THE COMPREHENSION
OF UML CLASS DIAGRAMS VIA EYE TRACKING

A thesis submitted

to Kent State University in partial

fulfillment of the requirements for the

degree of Master of Science

by

Shehnaaz Yusuf Patel Dawoodi

December, 2007

Thesis written by

Shehnaaz Yusuf Patel Dawoodi

B.S., Universiti Teknologi Malaysia, 1999

M.S., Kent State University, 2007

Approved by

_____Dr Jonathan I. Maletic_______________, Advisor

_____Dr Robert A. Walker________________, Chair, Department of Computer Science

_____Dr Jerry Feezel_____________________, Dean, College of Arts and Sciences

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I express my gratitude to many people who guided, encouraged, and helped me in finishing my degree. I could not have accomplished this important milestone without my wonderful and loving husband *Huzefa*. I thank my parents *Yusuf* and *Fatima*, brothers *Firoz* and *Shiraz*, and sisters *Mumtaz* and *Kaneez* for their constant support and believing in me.

I could not have asked for a better advisor and mentor than *Dr. Jonathan Maletic*. He was always available for his timely assistance and guidance. I also greatly appreciated the valuable suggestions and insightful comments from my committee members *Dr. Javed Khan* and *Dr. Arden Ruttan*. My research work would not have sailed smoothly without *Dr. David Robbins*, *Dr. Jason Holmes*, and *Mr. Aaron Rosenberg* generosity in allowing me to use their Tobii eye-tracker.

Finally, I formed a very unique memory of the Software Development Laboratory <SDML>. Working with the <SDML> members was truly educational and more importantly fun.

CHAPTER 1

Introduction

The use of eye tracking to complement traditional usability assessments (e.g., surveys and questionnaires) is gaining popularity in a variety of domains [Bojko 2005; Duchowski 2003; Hyona, Radach, Deubel 2003]. This development can be attributed to a number of recent advancements in the eye-tracking technology. High quality, extremely accurate, and user-friendly equipment is available today. These systems are relatively affordable and easy to use but their most noteworthy capability is the ability to collect a human subject's eye gazes in a non-obtrusive manner. This accurate data can then be used to help understand the cognitive process involved in the processing of visual data [Bednarik, Tukiainen 2006; Bojko 2005; Iqbal et al. 2005].

Pictorial representations such as the Unified Modeling Language (UML) Class diagrams [Booch, Rumbaugh, Jacobson 2005] are commonly used to model the design and structure of software systems. Representations, including layouts of UML class diagrams, are a general research topic with regards to software comprehension and maintenance activities. Investigations in the software visualization and program comprehension communities have primarily focused on effective layout schemes [Andriyevska et al. 2005; Eiglsperger, Kaufmann, Siebenhaller 2003; Sun, Wong 2005] and key aesthetics criteria [Eichelberger 2002; Eichelberger 2003; Gutwenger et al. 2003] with the goal of aiding and/or enhancing the human cognitive process. A number of usability studies have been reported in the software visualization research literature that evaluate UML class diagrams, including those with additional semantic information (e.g., class stereotypes), for an effective representation in addressing various software evolution tasks [Andriyevska et al. 2005; Arisholm et al. 2006; Kuzniarz, Staron, Wohlin 2004; Staron, Kuzniarz, Thurn 2005]. These studies typically form conjectures and/or draw conclusions from the data explicitly collected from subjects' via a combination of questionnaires, experience reports, and feedback comments after a designated task is completed. This raises a potential threat to the validity of the study namely: How well the subjects' responses on the completion of a task match the "reality" they observed while performing that task? For example, a subject may forget to report (or misreport) an observation after a lengthy task.

Here, we take a different approach to assess the representations of UML class diagrams. We use an eye-tracking equipment to implicitly collect a subject's activity data in a non-obtrusive way as (s)he is interacting with the diagram in performing a given task. The equipment collects three forms of pertinent data: 1) the eye-gazes with respect to the visual presentation; 2) an audio recording; and 3) a video recording of the subject during the session. We believe that eye tracking provides promising measures of cognitive workload and comprehension because it provides real-time information of the workload and activities, does not disrupt/distract a subject from focusing a task at hand, and is less intrusive than other traditional measures such as experience reports or users feedback. The next section describes several factors motivating our research.

1.1 Motivation

The main motivation behind our work is provide an additional supporting or alternative evidence as to how the human subjects comprehend the UML class diagrams. Prior work through various approaches (i.e., case study, questionnaire, feedback) has shown that certain aesthetic criteria (e.g., reducing edge crossings), layout strategies, and additional design information (e.g., class stereotypes) could aid in a better understanding of the UML class diagrams. In this work, we use an eye-tracking equipment to gather the real time eye-gaze data while the human subjects are carrying out a set of UML comprehension tasks. Our goal is to utilize these objective measures to approve or disapprove previous results.

1.2 Research Questions

The overarching research direction is to obtain a better understanding of how software developers explore, examine, and navigate the UML class diagrams [Yusuf, Kagdi, Maletic 2007]. Our hope is that this understanding will help us develop more effective UML class diagram layout mechanisms, and other supplementary notations and visual representations for software design information. In this work, the following specific questions are investigated:

- Which UML class diagram layout is the most effective for software comprehension and design tasks?
- Does the use of class stereotype information provide additional assistance?
- Is the use of colors to map semantic information on classes (entity, boundary, control) useful?

- What do people really look at (or not) in class diagrams?
- Is there an observable difference between experts and novices?
- What items in the diagrams do people fixate on the most?
- How do people navigate through the diagrams?

1.3 Organization of the Thesis

The remainder of the Thesis is organized as follows: Chapter 2 presents background on UML and eye tracking. Related work is presented in Chapter 3. Chapter 4 describes our study on assessing how people comprehend UML class diagrams. Our findings and analysis of the study are presented in Chapter 5. Chapter 6 concludes the Thesis emphasizing the contribution of this research on UML comprehension and the future work.

CHAPTER 2

UML and Eye Tracking Background

2.1 Introduction

This chapter introduces the UML class diagram notations and provides background on eye-tracking technology. Definitions of the basic eye movements used in eye tracking are presented. The use of these movements on a UML class diagram is also illustrated.

2.2 UML Class Diagram

The Unified Modeling Language (UML) is a standard way to model design of the software systems. In the UML, a class diagram shows the static structure of the system, namely the classes (an abstraction unit in object oriented analysis and design), their attributes/operations, and the relationships between the classes. A class in UML is drawn as a rectangle box split into up to three sections. The top section contains the name of the class, the middle section contains the attributes or data that the class contains, and the bottom section contains the operations (methods) that represent the behavior that the class exhibits, as shown in Figure 1. Class name may also contain stereotype information enclosed in the angle quotes <<*stereotype*>>, as shown in Figure 2. Stereotype is an extensibility mechanism in UML to denote that the class serves a special purpose (responsibility or intent). Following are the definition and description for each type of stereotypes used in our experiment [Overgaard, Palmkvist 2004]

- Entity: Entity classes models a concept or information managed inside the system. The instances of the entity classes are persistent and provide information to other instances in the system.
- Boundary: Boundary classes appear at the system boundary and handles communication between the interior of the system and the system's environment. Interaction between actors and system is handled by instances of boundary classes.
- Control: Control classes models a task or job to be performed in the system. It coordinates activity and act as glue between a boundary and an entity classes.

Classes collaborate with other classes using different types of structural and behavioral relationships. The most commonly found types of relationships in a UML class diagram are shown in Figure 3. Following are the definition and description for each type of a relationship [Rumbaugh, Booch, Jacobson 2004]

- Dependency: A dependency relationship between two classes indicates that one (independent) class is always needed to use the objects of the other (dependent) class, but the converse is not necessarily always true.
- Association: An association relationship between two classes means that a class contains a reference to the other class typically via an attribute.
- Aggregation: An aggregation relationship is used to model a part-whole relationship between classes. An object of a (whole) class consists of objects of other (part) classes. However, the same object of a part class can be shared among multiple whole objects.

- Composition: A composition relationship is similar to the aggregation relationship except that an object of a whole class owns (and not share) objects of the part classes.
- Generalization: A generalization relationship is use to model inheritance between the base (generalized) and derived (specialized) classes.

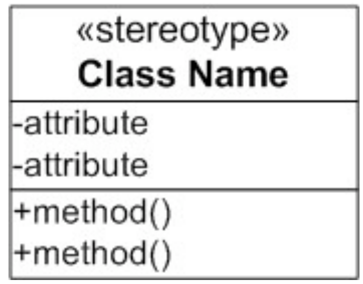

Figure 1. Notation for the class in the UML class diagram showing the compartments for attributes and methods

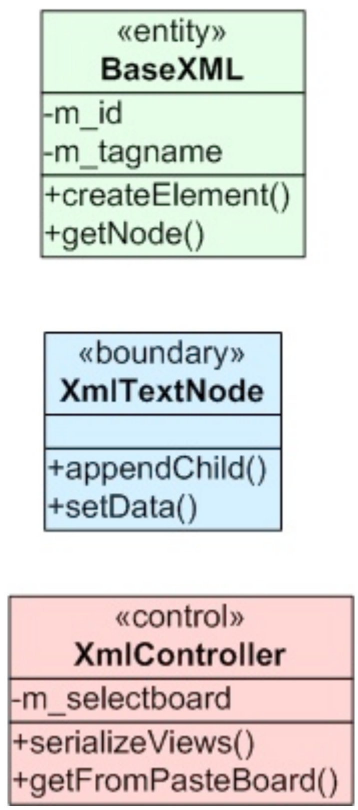

Figure 2. Entity, boundary and control stereotype shown on each class

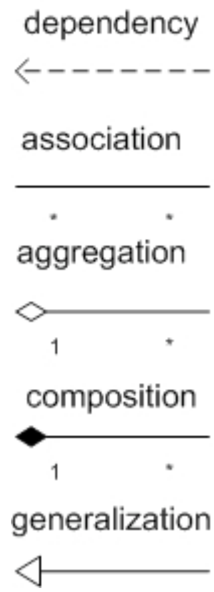

Figure 3. Notations for the different relationships in the UML class diagram.

2.3 Eye Tracking Equipment

The fundamental design of eye-tracking equipment is based on the physiology of the human visual capability [Duchowski 2003; Jacob 1990]. These systems use cameras to track eye movement. Specifically, we used a *Tobii 1750* eye-tracker (www.tobii.se) to capture eye movements and collect eye gaze data. Figure 4 shows the eye-tracker used in the experiment. In this equipment, the two cameras used to track the eye are built into a 17 inch flat-panel screen. Therefore, no restraints such as wearing a headband or goggles are placed on the human subject. This was not the case in older eye tracking equipment. This provides a normal computer-operating environment during the study. Moreover, the *Tobii 1750* eye-tracker is very accurate with an error rate of less than 0.5 degrees and a sampling rate of 50MHZ. Software that records the XY screen coordinates of eye gazes and supports analysis of eye movements is also provided along with the eye-tracker system. An audio/video recording is also made of each study session.

2.4 Fixations, Saccades and Scanpath

The underlying basis is to capture various types of eye movements that occur while humans physically gaze at an object of interest. Among these, fixation and saccade are the two most widely used eye movements in these types of studies. According to Sibert et al. [Sibert, Templeman, Jacob 2000], fixations and saccades are the two most important elements in the behavior of the eye in the HCI. They are most suitable for the analysis of visual search tasks and exploration of the visual environment.

Definition: Fixation is the stabilization of eyes on an object of interest for a certain period of time.

<u>*Definition*</u>*: Saccades* are quick movements of the eyes that move interest from one location to the next (i.e., refixates).

<u>*Definition*</u>*: Scanpath* is a directed path formed by saccades between fixations.

When a subject directs her/his gaze onto an object, the eyes move so that the image of the target object appears on a part of retina, and hence can be seen clearly. The general consensus in the eye tracking research community is that the processing of visualized information occurs during fixations, whereas, no such processing occurs during saccades [Hyona, Radach, Deubel 2003; Jacob 1990]. Humans used saccades to locate interesting parts in a visual scene to form a mental model. A saccade ends with a fixation, a moment of relative stability when the signals from the eyes are processed in the human brains for making sense of the information received.

Figure 5 shows the recording of eye positions superimposed on a UML class diagram. The numbered circles represent fixation and lines between them represent saccades. The size of a fixation (i.e., area of a circle) is proportional to its time duration. The numbering of circles represents the ordering of fixations. For example, in figure 1, the fixation labeled with the number 35 on the class *NTuple* happened before the fixation labeled 36 on the class *NTupleController.* That is, the class *NTuple* was looked at before the class *NTupleController*. The scanpath in this case is directed to the left and downwards. A big circle on the class *PyNTuple* shows that a large amount of time was spent on this class. The eye-tracker captures fixation and saccades in the form of XY coordinates of the visual screen (in this case a UML class diagram) so that we can determine what was being looked at in the visual presentation.

2.5 Eye Tracking and Human Cognition

The eye tracking methods generally rely upon the *eye-mind hypothesis*, which states that when looking/fixating at a visual display and performing a task, the location of one's gaze-point corresponds to the person's thought or cognitive process. This hypothesis only applies for a task that requires the encoding and processing of the visual information to achieve clearly specified goals/answers [Hyona, Radach, Deubel 2003]. In our assessment, we have designed tasks that present the problem to be investigated as well as ensuring that extraneous peripheral information (i.e., noise, poor lighting, distracting environment, blinking screen) are eliminated.

Figure 4. The 1750 Tobii eye-tracker (www.tobii.se).

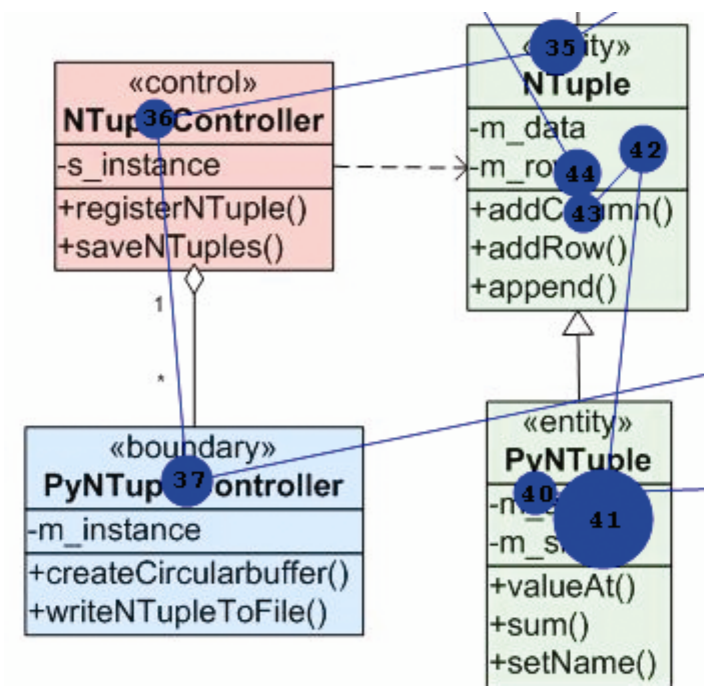

Figure 5. Gaze information on a UML Class Diagram. Fixations are represented with circles and saccades with lines connecting the circles.

2.6 Summary

The definitions of eye movements: fixations and saccades were discussed. The details on the eye tracking equipment used in the experiment were given. An overview of the UML class diagram was also discussed. The following chapter gives details on the related work relevant to our research.

CHAPTER 3

Related Work

3.1 Introduction

There are two research areas that are related to our work. We discuss representative works in the UML and eye-tracking usability studies.

3.2 UML Class Models

Sun et al. [Sun, Wong 2005] proposed key criteria and guidelines for the effective layouts of UML Class diagrams based on the perceptual theories. Authors' evaluated fourteen criteria (i.e., orthogonal, inheritance direction, color) on UML class diagrams and concluded that the perceptual factors are important for devising diagram design and guidelines. Kurniaz et al. [Kuzniarz, Staron, Wohlin 2004] and Staron et al. [Staron, Kuzniarz, Thurn 2005] evaluated the influential role of stereotypes in understanding UML class and collaboration diagrams. The result shows that the use of stereotypes plays a significant role in comprehension of the UML models by giving specific semantic properties to elements. Andriyevska et al. [Andriyevska et al. 2005] found that the layouts based on design and architectural information assist more in comprehension of UML class diagram than those solely based on the general graph drawing aesthetics. In their work, multiple clusters layout was found to be the most effective while the use of color on different element (control, boundary, entities) helps in understanding the UML Class Diagram by narrowing downs the search scope to one stereotype. Eichelberger

[Eichelberger 2002; Eichelberger 2003] proposed a set of aesthetic criteria and semantic clustering of nodes to increase the readability of UML class diagrams. The graph-drawing framework called SugiBib produces UML Class diagrams based on the design criteria (i.e., number of children, class size metrics), HCI criteria (Spatial distribution, enlargement) and principles of aesthetics (i.e., edges direction, semantic clustering of nodes).

Purchase et al. [Purchase et al. 2001; Purchase, Allder, Carrington 2002; Purchase et al. 2001] conducted user studies to evaluate the effect of aesthetics criteria (i.e., minimize bends, edge crossing, orthogonal) on the UML diagrams. The results from this experiment showed that a large number of subjects preferred diagrams with fewer crosses, bends and a more orthogonal drawing. Tilley et al. [Tilley, Huang 2003] investigated the use of UML syntax, semantics, spatial layout, and domain knowledge in system evolution tasks. Eiglsperger et al. [Eiglsperger, Kaufmann, Siebenhaller 2003] proposed an automatic layout algorithm for UML class diagrams. Their algorithm is based on the topology and shape metrics that try to minimize the edge crossing, bends, and occupied area.

Gutwenger et al. [Gutwenger et al. 2003] proposed a new approach for visualizing UML Class Diagram that follows certain aesthetic criteria in their tool GoVisual. The proposed technique (i.e., same direction for generalization, no nesting of hierarchies, use colors) is applied on an orthogonal diagram that features hierarchical and non-hierarchical elements in such a way that all the directed edges of a component follow the same direction. Musial et al. [Musial, Jacobs 2003] developed a fisheye view that displays less detail for components with a smaller degree of interest and apply selective aggregation techniques to hide components that are beyond a specified degree of interest. Dwyer [Dwyer 2001] applied the physical forces to the elements of the UML class diagram to lay out a balanced graph. The Force Directed Algorithm (FDA) proposed by the author is realized in the tool called Wilma. Result shows that 3D visualization of complex UML diagrams convey more information easily than 2D. Diagrams produced by this approach are less cluttered, with no intersections and with a minimal distance between connected components.

Briand et al. [Briand et al. 2005] conducted controlled experiments to understand the use of OCL (Object Constraint Language) in comprehension of a system's functionality, behavior, and structure based on the UML Model. They showed that the combined use of OCL and UML offers significant benefits in terms of defect detection, comprehension, and maintenance of UML analysis documents. Arisholm et al. [Arisholm et al. 2006] conducted a controlled experiments that investigates the impact of UML documentation on software maintenance, namely on the correctness and the effort of performing changes. Results showed that the UML documentation can provide significant improvements in the functional correctness of changes and overall quality of the design for complex tasks. Recently, Ricca et al. [Ricca et al. 2007] evaluated the effectiveness of Conallen's stereotypes (a type of web application notation) in improving the comprehension of web applications. In their experiments, subjects were provided with the source code and either a stereotyped or non-stereotyped UML diagrams. Their results shows that the novice subjects makes an extensive use of stereotypes while the experts relied more on a textual form of representation such as source code. The use of stereotypes is a single most factor that reduces performance gap between these two groups of people.

Kagdi et al. [Kagdi, Maletic 2007] presented a novel approach for focus+context views of UML class diagrams. The combined view provides the focus and details in UML notation while the context information is presented in the form of onion notations. Furthermore, the views preserve the structure and semantics of the class diagram as well as achieve edge reduction.

In one of the rare cases, Guehéneuc [Gueheneuc 2006] recently used eye-tracking to study the comprehension of the software engineers on the class diagrams. In the experiment, subjects were given two class diagrams for perusal and asked two questions while the eye gaze data were collected. An older version of a head-mounted eye-tracker was used which produces raw fixation data. Author aggregates fixations in respect to the area of the UML class diagram to find classes that received more or less attentions than others. The preliminary result reported by the author is the apparent lack of use of relationships among classes. However, their study was limited with regards to the questions and the overall scope. Additionally, they used a head mounted system that is quite intrusive and no more accurate than what we used.

3.3 Eye Tracking and Usability

Jacob [Jacob 1990] discusses the human factors and technical considerations in using eye tracking in Human Computer Interaction (HCI). Beymer et al. [Beymer, Russell 2005] developed the tool *WebGazeAnalyzer* to record and analyze eye gazes on web browsing sessions. Statistical analysis was done to measure information from fixations, such as time and speed of reading. Uwano et al. [Uwano et al. 2006] used eye tracking to characterize the individual's performance in reviewing source code. In their work, authors used fixation data to identify reviewers scan pattern. The quantitative analysis showed that reviewers who did not spend enough time for the scan tend to take more time for finding defects. They observed that the subjects were likely to first read the whole lines of the code from the top to the bottom briefly, and then to concentrate some particular portions.

Nakamichi et al. [Nakamichi et al. 2006] advocate the use of gaze-point velocity to detect the low usability web pages. Behavior used for the evaluation includes operation time (i.e, browsing time, task time), mouse movement (i.e, moving distance, click positions), and eye movement (i.e., moving distance, and moving speed). Khiat et al. [Khiat, Matsumoto, Ogasawara 2004] studied the relation between subjects understanding and their eye movements on the text in a non-native language. They showed that when users find unfamiliar word, they might fixate the word many times. The authors have discussed the importance of grounding user's gaze within its context and a difficulty rate is used to get better detection results. Authors [Law et al. 2004] conducted a study comparing the eye movements of expert surgeons and novices performing a computer-based surgery using the simulator. The results showed performance and eye movement differences between the expert and novice. The experts were quicker, and the results show a trend that they were more accurate than novices and committed fewer errors. Pan et al. [Pan et al. 2004] studied factors such as gender information, web page viewing order, and different types of website (news and shopping) by using eye tracking measures. Whalen et al. [Whalen, Inkpen 2005] conducted a study to determine the elements in web browsers that are viewed (and ignored), and how easily they can be noticed. Results demonstrated that the lock icon in the browser security cue is most often looked at while the certificate information is rarely used.

Bednarik et al. [Bednarik, Tukiainen 2006] applied eye tracking to study comprehension of Java programs using the Jeliot visualization tool. In their experiment, more experienced participants read the code first, and then run the program in a single execution. The visualization provided the experts with additional information, to confirm and fine-tune their previously established mental model and hypotheses. On the other hand, less experienced programmers did not read the code at the beginning, but instead animated the program several times and used the tool to visually explain the program execution. Other findings show that in the early phases of comprehension, visualization provides more important information and plays more important role than at the later stages.

Iqbal et al. [Iqbal et al. 2005] investigated the mental workload demanded by computer-based tasks perform by users in an eye tracking study. A user's subjective rating, task completion time for each task, as well as the user's pupil data (eye movement information) and on the screen activities were collected. Their results show that a more difficult task demands a longer processing time, induces higher subjective ratings of the mental workload, and reliably evokes a greater papillary response than a less difficult task.

3.4 Summary

This chapter briefly surveyed two main research fields within the eye tracking and software engineering area. In summary, our literature search revealed one study using the eye tracking equipment on the UML class diagrams. The differences between this study and our work were stated. We now describe the design of our study.

CHAPTER 4

Assessment Study

4.1 Introduction

The principal goal is to obtain an understanding of how human subjects use different types of information in UML class diagrams in performing their tasks. In a nutshell, human subjects were given specific tasks to perform on diagrams. An eye-tracker was used to capture their activities in terms of fixation, saccades, audio, and video. The following is a more detailed description of the various components of our study.

4.2 UML Class Diagram Layout

We used UML class diagrams representing the design of the open source *HippoDraw* software (www.slac.stanford.edu/grp/ek/hippodraw). *HippoDraw* is a statistical data analysis application that is primarily written in C++ and uses the Qt library for GUI. It also provides an API framework via a Python interface.

We used three different layout techniques of UML class diagrams for our investigation. Our selection of these layout methods are based on previous work in assessing layouts [Andriyevska et al. 2005]. These diagrams vary in layouts, semantic information (e.g., stereotype), and secondary notations (e.g., color).

Definition: The *orthogonal* layout focuses on the minimization of the edge crosses and bends. Multiples of 90 degree angles are used to position the intersecting edges [Eichelberger 2003; Eiglsperger, Kaufmann, Siebenhaller 2003; Purchase, Allder, Carrington 2002]. This layout is adopted from general graph drawing algorithms and is typically available in UML modeling and drawing tools.

<u>*Definition:*</u> The *three-cluster* layout positions classes into three clusters (i.e., boundary, control, and entity) based on their design or architectural roles. Classes that are stereotyped entities, in terms of UML vocabulary, are placed in a single cluster. Similarly, classes that are stereotyped boundary and control form the other two clusters. This is an example of layouts that use the general role of a class in the high-level design modeling and analysis of a software system via UML.

<u>*Definition:*</u> The *multiple-cluster* layout is a further specialization of the three-cluster layout. Related classes that are responsible for a specific functionality of a software system are positioned in a single cluster. This is an example of layouts that further use the responsibilities of classes in modeling, analysis, and realization of an application domain specific concept. For example, a cluster could map to a functional requirement of a system. Therefore, the number of clusters in a layout could be equivalent to the number of functional requirements.

Figure 6, Figure 7, and Figure 8 show examples of orthogonal, three-cluster, and multiple-cluster layouts for the same UML class model. Colors and textual annotations (i.e., *<<entity>>, <<control>>, and <<boundary>>*) are used to represent class stereotypes. Boundary, entity, and control classes are represented by three different colors (blue, green, and red colors in our study). The orthogonal layout does not use the semantic information such as stereotype of a class in positioning it on a diagram, whereas the other two do. However, we also made the stereotype information in the orthogonal layout with textual annotation so that all the diagrams exhibit the same design information.

4.3 Tasks

According to Renshaw et al. [Renshaw, Finlay, Webb 2006], the two most important issues when conducting any usability studies utilizing the eye tracking system are, defining a suitable task to evoke necessary eye movement that is being investigated and the use of eye tracking metrics as a means of providing quantitative measures. The tasks given to the subjects in our study consist of the subjects answering specific questions by reading/viewing UML class diagrams. We designed two types of questions, one set dealing with basics of UML class diagram and the other set related to the software design. The set of diagram questions deal with the characteristics of the classes, attributes, methods, relationships, and general notations. For example, what is the type of relationship between two given classes? This set of questions is aimed at understanding the user activities in performing general exploration, explanatory, and navigation tasks in a UML class diagram.

The set of software-design related questions are concerned with general software design understanding, extensibility, and changeability. For example, name the class that could be extended to accommodate a new GUI functionality. These questions are aimed at providing insight as to how software developers approach, process, and accomplish design tasks by utilizing UML class diagrams. The questions in this set were planned in such a way that it needed minimal knowledge of the finer design, implementation, and domain minutia of *HippoDraw*, and knowledge of fundamental software design principles to address them.

Table 1 shows the set of 12 UML questions and Table 2 shows the set of 15 software design questions used in our study. Table 3 shows the distribution of questions that are asked for the six modules of *HippoDraw* using the three different types of layouts. Only UML questions are allocated to the module *High-Level* and only software design questions are allocated to the modules *XmlNode* and *Canvas*. The remaining three modules *Python Wrappers*, *PlotterBase*, and *Tuple* are allocated questions from both sets. Notice that the same question is not asked for two different layouts. This was done to avoid any learning bias that may occur due to the same question asked twice. However, very similar questions were asked to give a fair coverage to all the layouts. We felt that this distribution allows us to analyze common and exclusive behavior of the three layouts in supporting two different types of tasks.

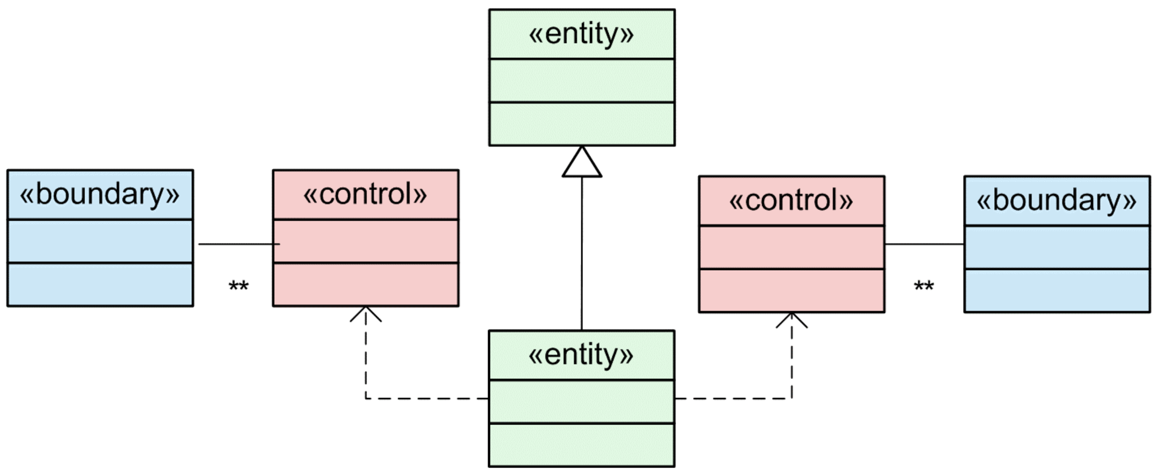

Figure 6. Orthogonal Layout

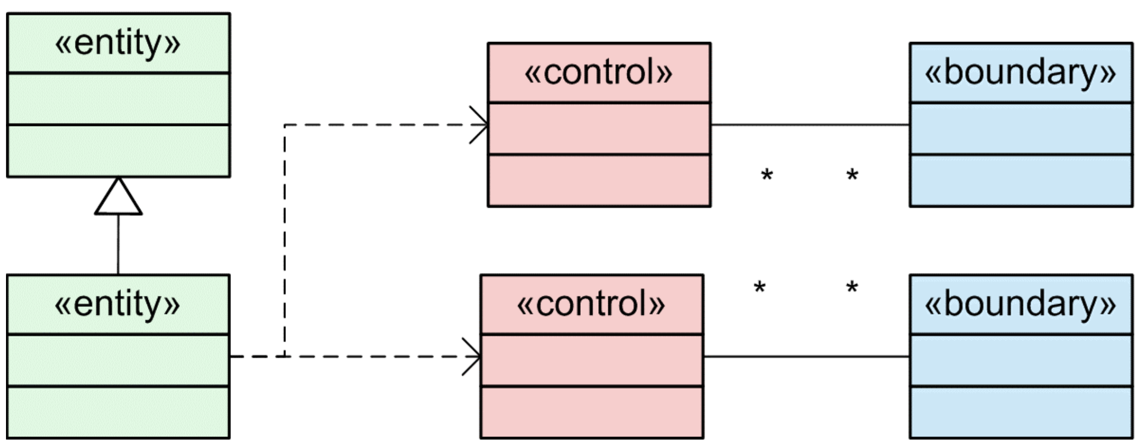

Figure 7. Three-Cluster Layout

Table 4 shows the number of classes in a UML class diagram that are used from the corresponding modules of *HippoDraw*. Overall 100 unique classes are used from the *Hippodraw* system. We selected six class models that represent six logical subsystems or a set of related functionalities. We manually engineered three class diagrams with orthogonal, three-cluster, and multiple-cluster layouts to represent each model. Each of the resultant 18 diagrams occupies approximately the same amount of physical screen space and consists of between 12 and 21 classes. The bound on the number of classes in a diagram is guided from Purchase's [Purchase, Allder, Carrington 2002] results on the optimal number of the classes beyond which there is a substantial cognitive overhead for comprehension tasks. Also, Sun et al. [Sun, Wong 2005] showed that a diagram with very dense information leads to difficulty in its readability. Therefore, our diagram shows only selective methods and attributes that are considered most relevant to the designated tasks. Further, we considered various advocated aesthetics criteria such as fewer edge bends and crosses, shorter edge lengths, and maximization of symmetry in the literature [Eichelberger 2002; Eichelberger 2003; Purchase, Allder, Carrington 2002].

4.4 Stimulus

Using the eye tracking terminology, an object that is viewed by a subject is known as the *stimulus*. We combine a question and the corresponding diagram into a single stimulus. The question or task description is placed in most cases at the top-left corner and the diagram occupied the remaining space. Research on the use of eye tracking for a variety of domains show a human bias for the top-left corner [Bojko 2005; Goldberg et al. 2002] and/or reading from the left to right [Beymer, Russell 2005; Khiat, Matsumoto, Ogasawara 2004]. In our experiment, we take the approach of centering the UML class diagrams and arrange the questions at the top-most diagram (left to right). Therefore, our chosen arrangement of the question and the diagram should help eliminate or drastically reduce this bias oriented noise. Figure 9 shows an example of a portion of a stimulus meeting our criteria. In our study, a total of 27 stimuli are formed from the combinations shown in Table 3.

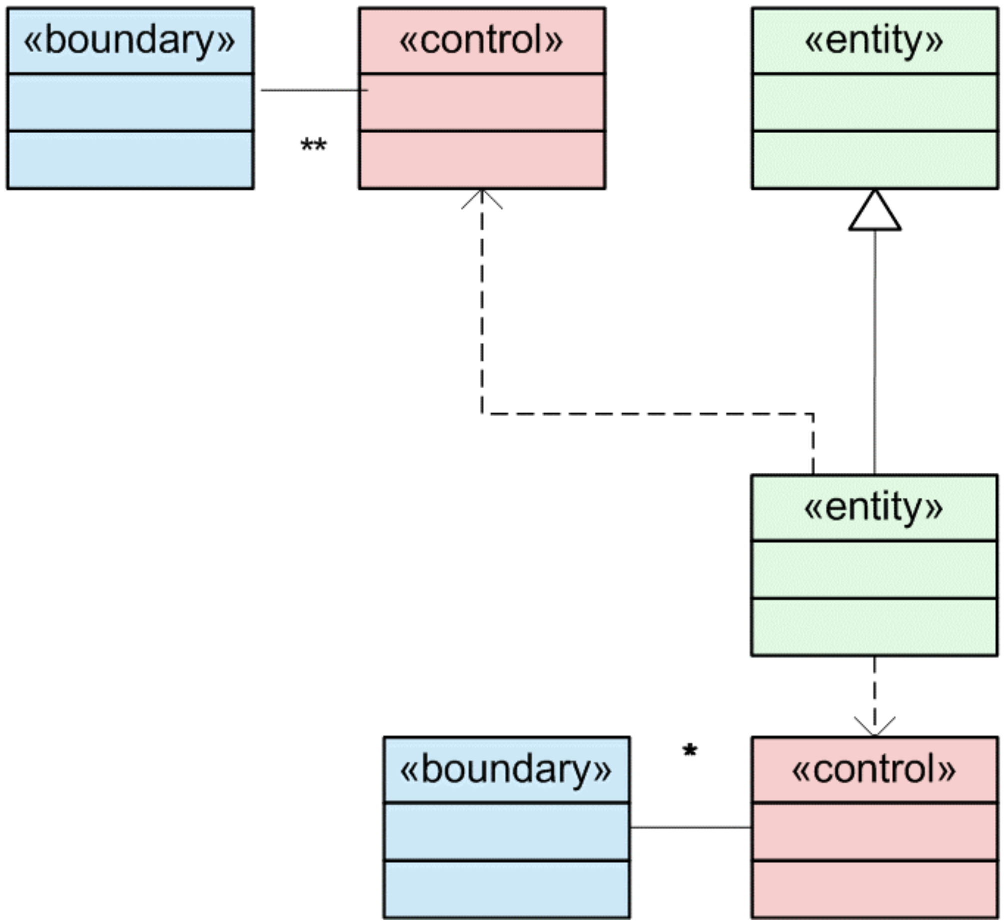

Figure 8. Multiple-Clusters Layout

Table 1. UML questions used in the study

No.	Questions
1	Identify the kind of relationship between class *ViewBase* and class *PlotterBase*.
2	Name the classes involved in aggregation.
3	Name the derived classes of the class *PlotterBase*.
4	Name the class with the method name *getAverage*.
5	Identify the kind of relationship between class *NTuple* and class *DataSource*.
6	Name all the classes involved in dependency.
7	Count the number of derived classes of the class *Observer*.
8	Name the class with the method name *objectiveValue*.
9	Identify the kind of relationship between class *DataSource* and class *Observable*.
10	Name all the classes involved in generalization.
11	Count all the classes involved in aggregation.
12	Name the class with the method name *registerNtuple*.

Table 2. Software design questions used in the study

No.	Questions
13	Name the class that a python wrapper uses to access data in the class *NTuple*.
14	Name the class responsible for managing XML serialization.
15	Name the class that controls the active window of an application.
16	Name the base class for axis representation hierarchy.
17	Name the class through which a boundary class could access data in the class *NTuple*.
18	Name the class that is a python wrapper for a class with the method name *adduct.*
19	Name the classes that are specialized for XML processing in QT.
20	Name the class that responds to the toolbar events from windows and messages sent by the class *Inspector*.
21	Name the class that plots point in 2D.
22	Name the class through which a boundary class could access data in the class *DataSource*.
23	Name the entity class that is responsible for storing data.
24	Name the entity class that could be extended to specify a new property (besides *Font* and *Color*) in XML
25	Name the concrete class that displays data in a tabular format.
26	Name the class that sets the range and scale of the axis.
27	Name the class that gets data from the class *DataSource* objects and uses functions from the class *FunctionBase*.

Table 3. Distribution of questions for the three UML class diagram layouts and their corresponding modules in HippoDraw software.

Modules	Orthogonal	Three-Cluster	Multiple-Cluster
High-Level	1	5	9
Python Wrappers	2, 13	6, 18	10, 23
PlotterBase	3, 16	7, 21	11, 26
Tuple	4, 17	8, 22	12, 27
XmlNode	14	19	24
Canvas	15	20	25

Table 4. Number of classes used from the design of corresponding HippoDraw software.

Modules	Number of Classes
High-Level	14
Python Wrappers	15
PlotterBase	21
Tuple	19
XmlNode	12
Canvas	19

4.4.1 The Subjects

Volunteers who had completed undergraduate and/or graduate level of software engineering coursework and used the UML class diagrams for academic and/or industry projects were used as subjects. We secured nine such subjects: three faculty, four doctoral students, one master student, and one undergraduate student. These subjects were all from computer science but had varying degrees of software design and programming experience.

Additionally, we had three non Computer Science graduate students who had no knowledge of UML and very little or no software development experience. We incorporated these subjects in the study to compare results from these two groups and see if there is any inherent difference in their eye movements.

4.5 Running the Study

The study consisted of subjects viewing the stimuli and verbally responding to the stated questions. The entire study was conducted over a two-day period. The subjects were informed well in advance of the schedule of their sessions. On the day of the study subjects were given a single page UML notation guide along with the introductory information of *HippoDraw*. Also the subjects were briefed on the eye-tracking equipment as to how it works and what information would be recorded. They were informed that the eye tracking system automatically records their audio, video, and eye movements on the class diagram.

All the subjects were given the 27 stimuli (comprehension tasks). Only one subject at a time performed the study and it took between 10 and 20 minutes to complete one session. The subject was stationed comfortably in front of the eye tracker at a distance of approximately 60 cm and the eye-tracker was calibrated for their individual use to verify that the system was working properly. This process takes less than a couple minutes to complete. After this initial step, the environment in front of them was just a common desktop Windows operating environment.

The subjects were then instructed to read the question on a stimulus loudly and verbally answer it so that they could be recorded. There was no time limit on individual stimulus or the entire session. After concluding the task on a stimulus, the subjects were asked to say “next” so that the auditor could make them transit to the next stimulus. The set of 12 UML questions stimuli was presented before the set of 15 software design questions stimuli. The auditor verbally warned the subjects of the transition from one set to the other. The same diagram was not presented in consecutive stimuli in order to avoid immediate learning bias occurring due to a mental picture in the short-term memory. The subjects were encouraged to verbally provide their observations, comments, and feedback during and after the study.

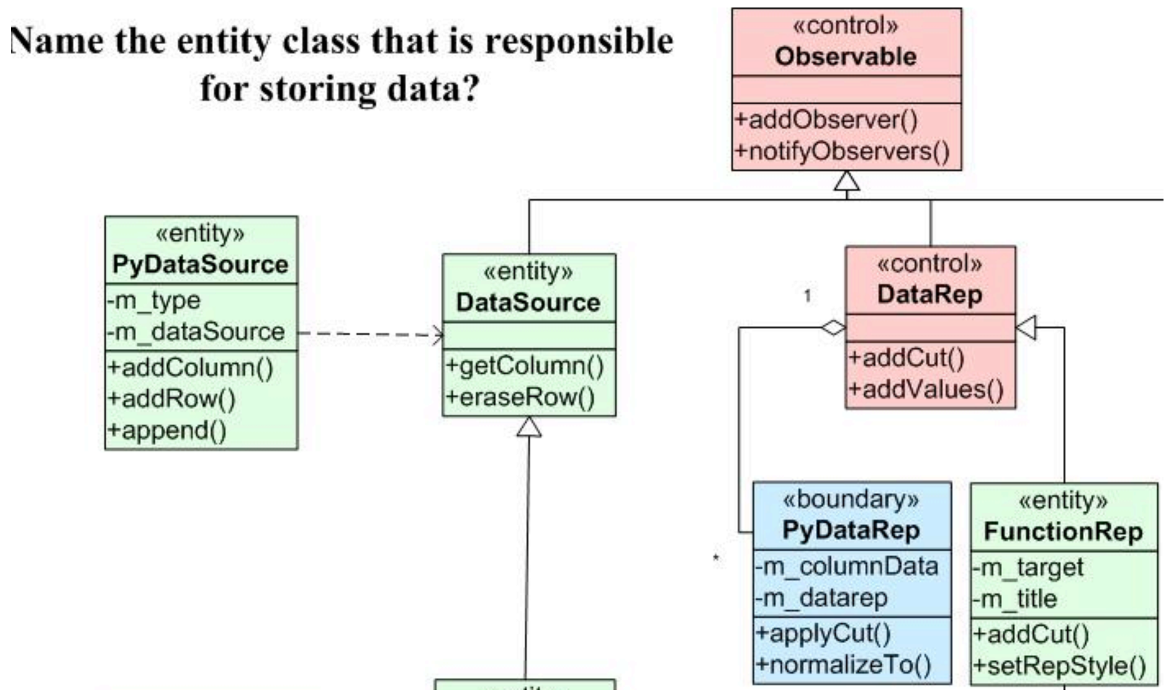

Figure 9. A portion of stimulus with a question in the top-left corner and the UML class diagram occupying the rest of the visual space.

CHAPTER 5

Analysis and Results

We discuss the analysis of the data collected from our study to obtain an understanding of subjects' visual activities in answering questions with the three layouts.

5.1 Subject and Question Classification

We analyzed the accuracy and response time of the answers to the 27 stimuli using the audio and video recordings of the experiments.

Figure 11 shows the number of correctly answered questions. The remainder of the questions were either incorrectly answered or skipped. Eight of the nine computer science subjects answered all the 12 UML questions correctly. No one answered all of the 15 design questions correctly.

The subjects with no UML knowledge prior to the study were able to answer a number of UML questions after reading the one-page description of the notation. Based on the performance of subjects in answering the questions, we classified them into the following groups:

- Both UML and design agnostic (UADA): Subjects that demonstrated very little knowledge of UML and software design. Three subjects (K, L, and J) are found in this category. These subjects took between 14 and 16.5 minutes each to complete the study.

- UML expert but inexperienced designer (UEDI): Subjects that seem very skillful in UML but seem to exhibit a lack of software design experience. Only one subject (A) is found in this category. This subject took approximately 13.9 minutes to complete the entire study.
- UML expert and knowledgeable designer (UEDK): Subjects that seem to be expert in UML and knowledgeable in software design. Three subjects (D, C, and H) are found in this category. These subjects took between 8.5 and 14 minutes to complete the entire study.
- Both UML and design expert (UEDE): Subjects that exhibited commendable knowledge on both UML and software design. Five subjects (B, E, G, I, and F) are found in this category. These subjects took between 6.5 and 11.5 minutes to complete the study.

It should also be noted that subjects had varying reading speeds. Some were very fast readers while others read slowly and carefully. This is one of the main reasons why we cannot compare performance based purely on the time to complete a particular task.

The classification of subjects shows that we have representatives with varying UML and software design skills. Also, our questions were effective enough to enable this classification and this information is used in further analysis presented in the following sections. We now classify the tasks based on the performance of subjects to gauge the difficulty level in answering the questions.

Figure 11 show that most subjects with the exception of the UADA group answered all the UML questions. Therefore we believe that the UML questions were quite easy to handle and are not classified further.

We classified the 15 design questions based on the distribution of subjects answering them correctly and excluded the UADA group from this analysis. Questions that were answered correctly by subjects in the ranges [0%, 25%), [25%, 70%), [70%, 80%), and [80%, 100%] were classified as *easy*, *intermediate*, *difficult*, and *challenging* respectively. Table 6 shows the specific design questions in the respective categories. No subject answered the question numbered 20 correctly. Other questions were correctly answered by at least one subject. Table 5 shows the exact breakdown of questions by subjects who either answered correctly or incorrectly.

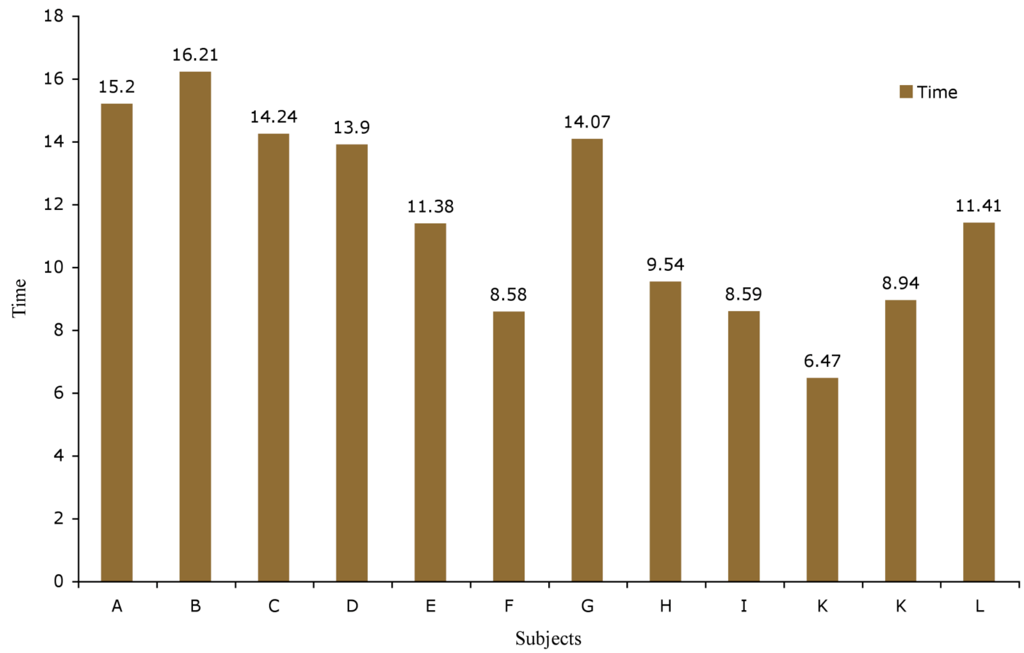

Figure 10. The time taken by all the subjects to complete the tasks.

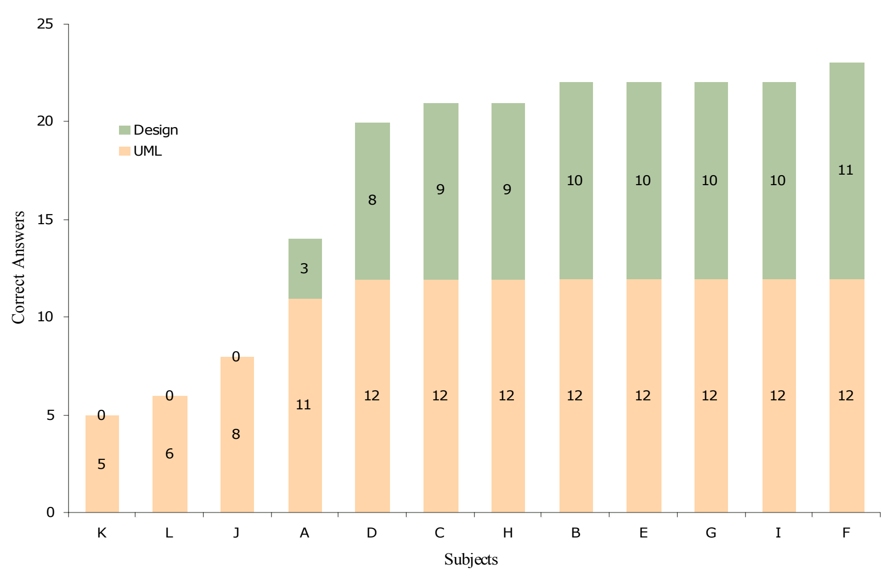

Figure 11. The number of correct answers for both UML and Design sets of questions

Table 5. Subjects who answered each design questions correctly or incorrectly.

No	Correct	Incorrect	Subject Corr	Subject Inc	Level	
20	0	A,B,C,D,E,F,G,H,I,J,K,L	0	12	Challenging	0
23	D	A,B,C,E,F,G,H,I,J,K,L	1	11	Challenging	11.11
18	B,E	A,C,D,F,G,H,I,J,K,L	2	10	Challenging	22.22
17	C,G,I	A,B,D,E,F,H,J,K,L	3	9	Difficult	33.33
26	C,E,F,H.	A,B,D,G,I,J,K,L	4	8	Difficult	44.44
25	B,E,F,G,I	A,C,D,H,J,K,L	5	7	Difficult	55.55
27	A,C,D,E,F,H	B,G,I,J,K,L	6	6	Difficult	66.66
13	B,C,E,F,G,H,I,	A,D,J,K,L	7	5	Intermediate	77.77
14	A,B,D,F,G,H,I,	C,E,J,K,L	7	5	Intermediate	77.77
22	B,D,E,F,G,H,I,	A,C,J,K,L	7	5	Intermediate	77.77
24	B,C,D,E,F,G,I,	A,H,J,K,L	7	5	Intermediate	77.77
15	B,C,D,E,F,G,H,I,	A,J,K,L	8	4	Easy	88.88
16	B,C,D,E,F,G,H,I,	A,J,K,L	8	4	Easy	88.88
19	B,C,E,F,G,H,I,	A,D,J,K,L	8	4	Easy	88.88
21	A,B,C,D,FG,H,I	E,J,K,L	8	4	Easy	88.88

Table 6. Classification of design questions based on the percentages of subjects correctly answering them. The question numbers correspond to the questions in Table 2

Level	Questions
Easy	15, 16, 19, 21
Intermediate	13, 14, 22, 24
Difficult	17, 25, 26, 27
Challenging	18, 20, 23

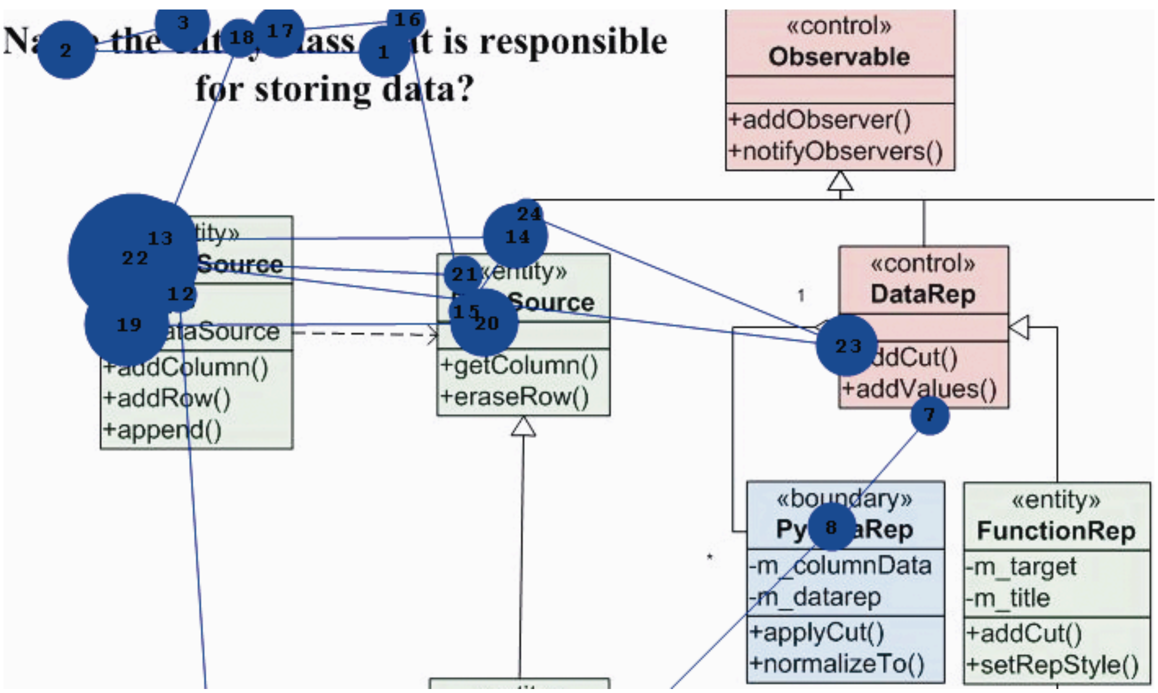

Figure 12. Gaze plot for a portion of the stimulus shown in Figure 9

5.2 Exploration, Examination, and Navigation

Here, we focus on trying to understand how subjects use their eye movements for:

- Exploration of visual space: How they perform searches on the UML class diagram to locate objects required for a given task.
- Examination of visual objects: How they visualize, in detail, whole or parts of classes and relationships while accomplishing a given task.
- Navigation: How they move from one object of interest to the next after their discovery.

Gaze plots, such as shown in Figure 12, that provide fixations, saccades, and scanpaths are used in this analysis. We found the following:

- The eye-tracker captured the fixations at the granularities of class, attribute, and method textual names in the diagram. Most subjects directly explored only the part of the diagram that contained the names specified in the questions. For example, when a class containing a specific method name X was required, subjects only searched the parts of the class containing methods.
- A wide majority of the fixations are found on classes and relationships, and very few on the empty spaces.
- The first fixations were found only on the end of relationship symbols (e.g., diamond edge for aggregation) for questions regarding or involving relationships. Therefore, subjects start examining from the relationship-ends for answering specific questions about them. Only saccades were found on the rest of a

relationship symbol (i.e., the lines). So the line parts of relationship notations are used only for navigation purposes.

- All subjects in the UEDE and UEDK groups start exploring the diagrams from the center and moved towards the periphery.
- Subjects in the UADA and UEDI groups explore the diagrams from top-to-bottom, and left-to-right.

5.3 Stereotype Usage

Here, we discuss the use of explicit stereotype information that was provided in the form of textual annotations and color in the diagrams. Gaze plots and video recording were used to facilitate the analysis. We found the following:

- All subjects in the UEDE group and majority of the subjects in the UEDK group visually examined the textual annotations used for stereotypes in answering the majority of the design questions. This was evident by the number and size of the fixations on text.
- All subjects in the UEDE group and majority of the subjects in the UEDK group used the distinct class colors indicating their stereotypes to facilitate exploration and navigation through the diagrams.
- None of the subjects in the UADA and UEDI groups used the stereotype textual annotations and colors. Since they did not use this information they explored and examined almost all classes in the diagram.
- Subjects in the UEDE group divided the visual space of the UML diagram into clusters based on the stereotype color information. They used clusters as units of

navigation (and not classes). They narrowed down their search to the cluster potentially containing the answer and examined that cluster in detail.

- Subjects that used the above strategy answered questions correctly and quickly than others.
- When answering questions that involved both stereotypes and relationships, majority of the subjects in the UEDE group used stereotype to narrow down to the possible solution, and then located the appropriate relationship to complete the answer.

Also, we analyzed all the heatmaps consisting of cumulative fixations of all the subjects for a particular stimulus (i.e., task) and found support for all of the above findings. For example, the heatmap in Figure 13 for question 23 shows a large number of fixations on the textual annotations of stereotypes.

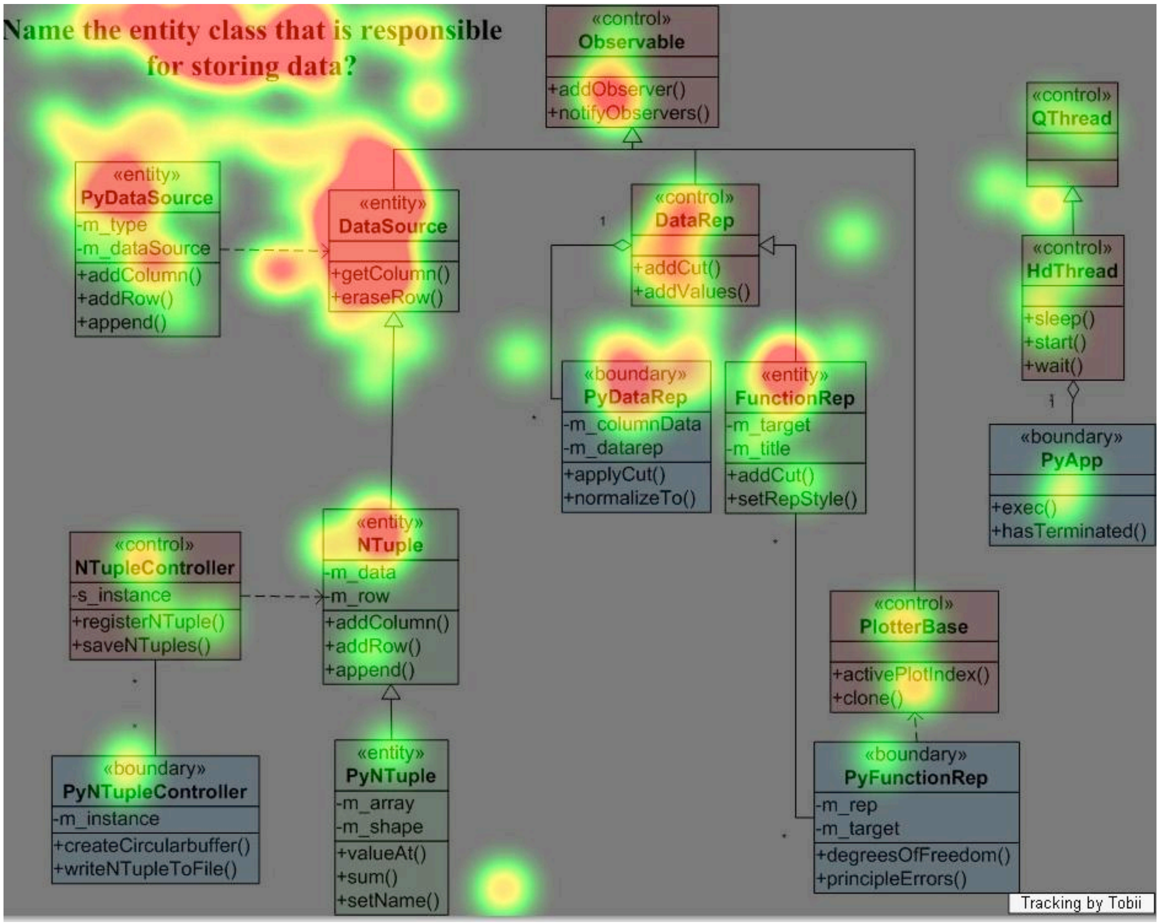

Figure 13. A heatmap showing the cumulative fixations of subjects on a specific stimulus. The colors red, orange, yellow and green indicate the decrease in number of fixations from highest to lowest. Best viewed in color.

5.4 Efficient Layouts

There is a wide variety of eye tracking metrics in the literature [Hyona, Radach, Deubel 2003]. The most frequently used metrics is the number of fixations. A large number of fixations is an indicator of the poor arrangements of objects in a stimulus. The determination of the efficiency of a layout is based on the total number of fixations on a stimulus. In our study, each stimulus corresponds to a diagram with one of the three layouts. Fewer total number of fixations on a stimulus means that the subject needs less effort to answer the associated question. We conjecture that if the total number of fixations is high then the classes and relationships are laid out in a way that leads to an inefficient visual exploration, explanation, and navigation. Such poor arrangement spans the attention of the subject across a number of objects instead of systematically narrowing down the visual space to only the relevant area of interest. Similar measures are used to assess the arrangement of objects in a visual environment in other domains that use eye-tracking methodology for assessment [Goldberg et al. 2002; Iqbal et al. 2005; Khiat, Matsumoto, Ogasawara 2004; Pan et al. 2004; Uwano et al. 2006].

The average number of fixations for a specific question is computed from the fixations of all the subjects (excluding the group UADA) on the associated stimulus. The column Average Fixations in Table 7 shows the average fixations for all the UML and design questions used in our study. In order to determine the relative effort required in answering the questions, four categories *low*, *intermediate*, *high*, and *extreme* are formed from the analysis of the average number of fixations. The median of all the average number of fixations of the stimuli is 34.33. The stimuli with average number of fixations in the range [0, 34), [34, 42), [42, 50), [50, 67) are classified as low, intermediate, high, and extreme respectively. The classification of the questions based on the accuracy of answers from Table 6 is also shown in Table 7 to facilitate comparison of difficulty level and the required effort.

Table 7. Classification of effort required to answer questions based on the average fixations taken over the number of expert subjects for each stimulus. The table is ordered by effort/average fixation. The column Levels are from Table 6.

Stimuli	Average Fixations	Effort	Levels
5	23.00	Low	Easy
23	23.56	Low	Challenging
11	24.67	Low	Easy
26	25.22	Low	Difficult
15	27.67	Low	Easy
8	28.00	Low	Easy
12	29.56	Low	Easy
6	29.89	Low	Easy
7	30.22	Low	Easy
18	30.56	Low	Challenging
9	31.22	Low	Easy
3	32.00	Low	Easy
19	32.56	Low	Easy
14	34.33	Medium	Intermediate
1	36.22	Medium	Easy
21	38.00	Medium	Easy
2	40.56	Medium	Easy
4	41.00	Medium	Easy
16	42.56	High	Easy
24	42.56	High	Intermediate
25	42.78	High	Difficult
10	43.56	High	Easy
22	44.22	High	Intermediate
13	62.44	Extreme	Intermediate
20	63.67	Extreme	Challenging
27	65.22	Extreme	Difficult
17	66.33	Extreme	Difficult

In order to compare the three types of layouts we compared the level of question and the effort needed in answering them. The baseline of comparison is that the level and effort should be directly related. That is, easy questions should require low effort, intermediate questions should require medium effort, difficult questions should require high effort, and challenging questions should require extreme effort. We refer to such questions having this property as *equal-effort*. Also, questions that require more effort than the corresponding baseline level are referred as *more-effort*, whereas those that require less effort than the corresponding baseline level are referred as *less-effort*.

Using Table 3 and Table 7 we can map questions and stimuli to the corresponding layouts. Table 8 shows that the multiple-cluster layout supports the highest number of questions at the equal-effort and the orthogonal layout supports the lowest number of questions at the equal-effort. Similar performance is seen in favor of multiple-cluster and three-cluster layouts, and against the orthogonal layout with respect to the more-effort category. Moreover, multiple-cluster and three-cluster layouts show support at less-effort, whereas no such support is found in the orthogonal layout. Clearly, the multiple-cluster layout outperforms the other two layouts, and the orthogonal layout is outperformed by the other two layouts, for both sets of UML and design questions.

Table 8. Distribution of questions based on level and effort. The multiple-cluster layout outperforms the others with respect to effort.

Layout Types	Equal-Effort	More-Effort	Less-Effort
Orthogonal	3	6	0
Three-cluster	4	4	1
Multiple-cluster	5	2	2

5.5 Discussion

In this section, we discuss a number of issues that emerged from the study and provide some additional context to the experiment results.

We observed that novices and experts have a different style of exploring the class diagram to complete the task. Novice subjects traverses the diagrams using the "trial and error" approach from top-to-bottom and left-to-right while expert subjects preferred a more structured approach. Their eyes move from the center of the diagram to the periphery in the "divide-and-conquer" manner. Experts utilized the stereotype information, and therefore providing an explanation as to why they were able to solve the given task more effectively and efficiently. Our results on the usefulness of stereotypes in navigating the UML class diagrams corroborate with the recent findings from other studies [Andriyevska et al. 2005; Staron, Kuzniarz, Thurn 2005].

The experts divided the visual space of the UML diagram into clusters and used clusters as units of navigation to narrow down the search. Interestingly, even when the layout is orthogonal, the experts were found to be dividing the visual space according to the stereotype information. Color is used to navigate from one corner to the other. Our results confirms the UML aesthetics criteria proposed by Eichelberger [Eichelberger 2002]. He advocates clustering of the nodes based on the semantic reasons (i.e., package membership and composition notation) and placement in the close vicinity to each other.

An issue that also needs further investigation is the similarity of the notational symbols in the UML class diagrams, such as the hollow diamond for the aggregation and the filled diamond for the composition relationships. The eye-tracking data analysis from our study shows that when a given tasks dealt with either an aggregation or composition relationship, subjects appeared being not able to discriminate or differentiate easily between them as both these relationships were examined equally even if the question required only one of them.

Finally, our experiment shows that the rectangular boxes dominantly attract more visual attention compared to the line portion (relationships) of the UML class diagrams. This results corroborates with the preliminary findings by Guehénéuc [Gueheneuc 2006], he reported the apparent lack of use of relationships among classes. The findings may provide a new insight and directions to researches [Kagdi, Maletic 2007] that focuses on reducing/minimizing lines and crossings in the UML class diagrams.

5.6 Threats to Validity

We discuss the internal and external validities of our approach with regards to the results obtained from its evaluation.

Internal validity refers to addressing the possible factors in our evaluation that bias the results one-way or the other and as such do not represent reality. All our subjects were from academia and volunteers. This raises the threat that they may not have been motivated enough to perform to their fullest capability and interest. Also, some subjects may have apriori knowledge of the *Hippodraw* system. We believe that this was less of an issue as there are no UML design documents publicly available (with the exception of *Doxygen* documents). The number of subjects (12) in our study may appear to be low, however, this range is typically found in eye-tracking studies [Bednarik, Tukiainen 2006; Goldberg et al. 2002; Iqbal et al. 2005].

External validity refers to addressing the general applicability of our approach and conclusions to any given dataset. We assessed UML diagrams with subjects from academia, 27 questions, 3 layouts, and one system. We tried to take adequate measures so that our study represents commonly found comprehension and design scenarios, however we do not claim that our results will generalize to any arbitrary task, layout, system, and subject combination.

5.7 Summary

The analysis of the experiments was presented. Results indicate subjects have a variation in the eye movements (i.e., how the subjects navigate the diagram) depending on their UML expertise and software-design ability to solve the given task. Layouts with additional semantic information about the design were found to be most effective and the use of class stereotypes seems to play a substantial role in comprehension of these diagrams.

CHAPTER 6

Conclusions and Future Work

This work, along with the work by Guehéneuc [Gueheneuc 2006] are the first studies to use eye-tracking equipment to assess how people comprehend UML class diagrams in the context of software design problems. The advent of new eye-tracking technology makes the use of this equipment easier and unobtrusive. This method of data acquisition is implicit and more objective compared to traditional usability study methods. It also opens the door for the creation of objective assessment metrics of class diagram layout.

Our findings showed that experts tend to use such things as stereotype information, coloring, and layout to facilitate more efficient exploration and navigation of class diagrams. Additionally, experts tend to navigate/explore from the center of the diagram to the edges whereas novices tend to navigate/explore from top-to-bottom and left-to-right.

We made some observations that need further investigation. Even if subjects could not answer the question correctly, they got very close to the answer by using stereotype and color information. Defining standards for the use of this type of additional information could lead to more readable and effective diagrams. Also, we observed that the close similarity in the notations for generalization and aggregation relationships could cause undue effort to differentiate. Using less similar visual notations may reduce the effort to understand diagrams.

6.1 Future Work

In the future, this work will be extended in the following three directions. Conduct the study with a larger number of subjects, various levels of UML expertise, and using another set of more comprehensive UML tasks. We plan to extend our study to utilize other open source system such as JEdit. Also, we propose using eye tracking equipment on other type of UML diagrams such as sequence and collaboration diagrams.

6.2 Summary

The research addresses the questions of how human subject comprehends the UML class diagrams. The eye tracking system is used to captures eye movement data of subjects performing UML comprehension task in a controlled experiment. Results obtained from the experiment not only validate prior work in the research area but also disclose new facts especially the differences between novice and expert navigating the UML class diagrams.

References

[Andriyevska et al. 2005] Andriyevska, O., Dragan, N., Simoes, B., and Maletic, J. I., (2005), "Evaluating UML Class Diagram Layout based on Architectural Importance", in Proceedings of 3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT'05), Budapest, Hungary, September 25th, pp. 14-19.

[Arisholm et al. 2006] Arisholm, E., Briand, L. C., Hove, S. E., and Labiche, Y., (2006), "The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation", IEEE Transactions on Software Engineering (TSE), vol. 32, no. 6, pp. 365.

[Bednarik, Tukiainen 2006] Bednarik, R. and Tukiainen, M., (2006), "An eye-tracking methodology for characterizing program comprehension processes", in Proceedings of 2006 symposium on Eye tracking research \& applications (ETRA), San Diego, California, pp. 125-132.

[Beymer, Russell 2005] Beymer, D. and Russell, D. M., (2005), "WebGazeAnalyzer: a system for capturing and analyzing web reading behavior using eye gaze", in Proceedings of CHI '05 extended abstracts on Human factors in computing systems, Portland, OR, USA, pp. 1913-1916.

[Bojko 2005] Bojko, A., (2005), "Eye Tracking in User Experience Testing: How to Make the Most of It. " in Proceedings of 14th Annual Conference of the Usability Professionals Association (UPA), Montréal, Canada.

[Booch, Rumbaugh, Jacobson 2005] Booch, G., Rumbaugh, J., and Jacobson, I.,(2005),Unified Modeling Language User Guide, 2nd ed., NJ, Addison Wesley.

[Briand et al. 2005] Briand, L. C., Labiche, Y., Penta, M. D., and Yan-Bondoc, H., (2005), "An Experimental Investigation of Formality in UML-Based Development", IEEE Transactions on Software Enginering (TSE), vol. 31, no. 10, pp. 833.

[Duchowski 2003] Duchowski, A. T.,(2003),Eye Tracking Methodology: Theory and Practice, London, Springer-Verlag.

[Dwyer 2001] Dwyer, T., (2001), "Three dimensional UML using force directed layout", in Proceedings of Australian symposium on Information visualisation - Volume 9, Sydney, Australia, pp. 77-85.

[Eichelberger 2002] Eichelberger, H., (2002), "Aesthetics of class diagrams", in Proceedings of First IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT), Washington, DC, USA, pp. 23-31.

[Eichelberger 2003] Eichelberger, H., (2003), "Nice Class Diagrams Admit Good Design?" in Proceedings of SOFTVIS'03, pp. 159-167.

[Eiglsperger, Kaufmann, Siebenhaller 2003] Eiglsperger, M., Kaufmann, M., and Siebenhaller, M., (2003), "A topology-shape-metrics approach for the automatic layout of UML class diagrams", in Proceedings of 2003 ACM symposium on Software visualization, San Diego, California, pp. 189-ff.

[Goldberg et al. 2002] Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., and Wichansky, A. M., (2002), "Eye tracking in web search tasks: design implications", in Proceedings of 2002 symposium on Eye tracking research & applications (ETRA), New Orleans, Louisiana, pp. 51-58.

[Gueheneuc 2006] Gueheneuc, Y. G., (2006), "TAUPE: towards understanding program comprehension", in Proceedings of conference of the Center for Advanced Studies on Collaborative research (CASCON), Canada.

[Gutwenger et al. 2003] Gutwenger, C., Junger, M., Klein, K., Kupke, J., Leipert, S., and Mutzel, P., (2003), "A new approach for visualizing UML class diagrams", in Proceedings of 2003 ACM symposium on Software visualization, San Diego, California, pp. 179-188.

[Hyona, Radach, Deubel 2003] Hyona, J., Radach, R., and Deubel, H.,(2003),The Mind's Eye:Cognitive and Applied Aspects of Eye Movement Research, 0-444-51020-6 ed., Amsterdam.

[Iqbal et al. 2005] Iqbal, S. T., Adamczyk, P. D., Zheng, X. S., and Bailey, B. P., (2005), "Towards an index of opportunity: understanding changes in mental workload during task execution", in Proceedings of SIGCHI conference on Human factors in computing systems, Portland, Oregon, USA, pp. 311-320.

[Jacob 1990] Jacob, R. J. K., (1990), "What you look at is what you get: eye movement-based interaction techniques", in Proceedings of SIGCHI conference on Human factors in computing systems: Empowering people, Seattle, Washington, United States, pp. 11-18.

[Kagdi, Maletic 2007] Kagdi, H. and Maletic, J. I., (2007), "Onion Graphs for Focus+Context Views of UML Class Diagrams", in Proceedings of IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 07), Banff, Canada, pp. 80-87.

[Khiat, Matsumoto, Ogasawara 2004] Khiat, A., Matsumoto, Y., and Ogasawara, T., (2004), "Task Specific Eye Movements Understanding for a Gaze-Sensitive Dictionary", in Proceedings of Conference on Intelligent User Interfaces, Funchal, Madeira, Portugal, pp. 265-267.

[Kuzniarz, Staron, Wohlin 2004] Kuzniarz, L., Staron, M., and Wohlin, C., (2004), "An Empirical Study on Using Stereotypes to Improve Understanding of UML Models", in Proceedings of 12th International Workshop on Program Comprehension (IWPC), pp. 14.

[Law et al. 2004] Law, B., Atkins, M. S., Kirkpatrick, A. E., and Lomax, A., J., (2004), "Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment", in Proceedings of Proceedings of the 2004 symposium on Eye tracking research \& applications, San Antonio, Texas, pp. 41-48.

[Musial, Jacobs 2003] Musial, B. and Jacobs, T., (2003), "Application of focus + context to UML", in Proceedings of Australian symposium on Information visualisation, Adelaide, Australia, pp. 75-80.

[Nakamichi et al. 2006] Nakamichi, N., Shima, K., Sakai, M., and KenIchi, M., (2006), "Detecting low usability web pages using quantitative data of users' behavior", in Proceedings of 28th International Conference on Software Engineering (ICSE'06), Shanghai, China, pp. 569-576.

[Overgaard, Palmkvist 2004] Overgaard, G. and Palmkvist, K.,(2004),Use Cases: Patterns and Blueprints, Addison Wesley.

[Pan et al. 2004] Pan, B., Hembrooke, H., Gay, G., Granka, L., Feusner, M., and Newman, J., (2004), "Determinants of web page viewing behavior: An eye-tracking study", in Proceedings of Eye tracking research & applications symposium on Eye tracking research & applications, San Antonio, Texas.

[Purchase et al. 2001] Purchase, H., C., Colpoys, L., McGill, M., Carrington, D., and Britton, C., (2001), "UML class diagram syntax: an empirical study of comprehension", in Proceedings of 2001 Asia-Pacific symposium on Information visualisation - Volume 9, Sydney, Australia, pp. 113-120.

[Purchase, Allder, Carrington 2002] Purchase, H. C., Allder, J.-A., and Carrington, D. A., (2002), "Graph Layout Aesthetics in UML Diagrams: User Preferences", J. Graph Algorithms Application, vol. 6, no. 3, pp. 255-279.

[Purchase et al. 2001] Purchase, H. C., McGill, M., Colpoys, L., and Carrington, D., (2001), "Graph drawing aesthetics and the comprehension of uml class diagrams: an empirical study", in Proceedings of Australian symposium on Information visualization, pp. 129-137.

[Renshaw, Finlay, Webb 2006] Renshaw, J. A., Finlay, J., and Webb, N., (2006), "Getting a measure of satisfaction from eyetracking in practice", in Proceedings of CHI '06 extended abstracts on Human factors in computing systems, Montreal, Quebec, Canada, pp. 1723-1726.

[Ricca et al. 2007] Ricca, F., Penta, M. D., Torchiano, M., Tonella, P., and Ceccato, M., (2007), "The Role of Experience and Ability in Comprehension Tasks Supported by UML Stereotypes", in Proceedings of Proceedings of the 29th International Conference on Software Engineering, pp. 375-384.

[Rumbaugh, Booch, Jacobson 2004] Rumbaugh, J., Booch, G., and Jacobson, I.,(2004),Unified Modeling Language Reference Manual, 2nd ed., Addison Wesley.

[Sibert, Templeman, Jacob 2000] Sibert, L. E., Templeman, J. N., and Jacob, R. J. K., (2000), "Evaluation and Analysis of Eye Gaze Interaction." in NRL Report. Washington, D.C.: Naval Research Laboratory.

[Staron, Kuzniarz, Thurn 2005] Staron, M., Kuzniarz, L., and Thurn, C., (2005), "An empirical assessment of using stereotypes to improve reading techniques in software inspections", in Proceedings of Proceedings of the third workshop on Software quality, St. Louis, Missouri, pp. 1-7.

[Sun, Wong 2005] Sun, D. and Wong, K., (2005), "On Evaluating the Layout of UML Class Diagrams for Program Comprehension", in Proceedings of 13th IEEE International Workshop on Program Comprehension, St. Louis, Missouri, USA, pp. 317-328.

[Tilley, Huang 2003] Tilley, S. and Huang, S., (2003), "A qualitative assessment of the efficacy of UML diagrams as a form of graphical documentation in aiding program understanding", in Proceedings of 21st annual international conference on Documentation (SIGDOC), San Francisco, CA, USA, pp. 184-191.

[Uwano et al. 2006] Uwano, H., Nakamura, M., Monden, A., and Matsumoto, K., (2006), "Analyzing individual performance of source code review using reviewers' eye movement", in Proceedings of 2006 symposium on Eye tracking research & applications (ETRA), San Diego, California, pp. 133-140.

[Whalen, Inkpen 2005] Whalen, T. and Inkpen, K. M., (2005), "Gathering evidence: use of visual security cues in web browsers", in Proceedings of 2005 conference on Graphics interface, Victoria, British Columbia, pp. 137-144.

[Yusuf, Kagdi, Maletic 2007] Yusuf, S., Kagdi, H., and Maletic, J. I., (2007), "Assessing the Comprehension of UML Class Diagrams via Eye Tracking", in Proceedings of IEEE International Conference on Program Comprehension (ICPC '07), Banff, Canada, 15th 2007, pp. 113-122.