Master Thesis

August 10, 2014

Using Visualization to Discover Patterns in Biometric Data

Visualization and Analysis of EDA, EEG and Eye-tracking Data Recorded from Developers Working on Code Comprehension Tasks

Hannes Tresch

of Göschenen UR, Switzerland (08-715-534)

supervised by

Prof. Dr. Thomas Fritz

Sebastian Müller

University of Zurich UZH

s.e.a.l. software evolution & architecture lab

Master Thesis

Using Visualization to Discover Patterns in Biometric Data

Visualization and Analysis of EDA, EEG and Eye-tracking Data Recorded from Developers Working on Code Comprehension Tasks

Hannes Tresch

University of Zurich[UZH]

s.e.a.l. software evolution & architecture lab

Master Thesis
Author: Hannes Tresch, h.tresch@gmail.com
Project period: February 12, 2014 - August 12, 2014

Software Evolution & Architecture Lab
Department of Informatics, University of Zurich

Acknowledgements

I would like to thank Prof. Dr. Thomas Fritz giving me the opportunity to write my master thesis at the software evolution and architecture lab at the University of Zurich. Moreover, I want to thank Sebastian Müller for the great assistance.

Abstract

The aim of this thesis has been to develop a visualization application that supports the analysis of psycho-physiological data recorded from software developers working on code comprehension tasks. Since psycho-physiological (also known as biometric) measurements, such as electrodermal activity (EDA), or electroencephalography (EEG), can be used as indicators for cognitive states, these data are used to detect the difficulties a developer might have experienced during source code comprehension activities.

Using three different visualization approaches, a dataset consisting of EEG, EDA and eye-tracking data was investigated with a special focus on the identification of various kinds of patterns (*e.g.*, code reading patterns or time-related biometric patterns). Based on the results of visual inspection (enabled by the visualizations developed for this purpose), significance tests were conducted to verify the findings. The analysis has confirmed eye movement patterns that had been described in previous research. New insights are made regarding time-related patterns. Metric correlations within task data were found, like the accumulation of eyeblinks with concomitant low values for mental focus (Attention eSense™ values). Additionally, metric differences in task pairs of inherently easier/harder tasks were analyzed and interpreted. For example, it was found that the use of mnemonic variable names instead of generic variable names result in a significant lower number of occurrences of the retrace declaration pattern.

Learning how software developers read source code and how they emotionally and physiologically react on reading code comprehension tasks could help to locate problematic code segments. Findings in this research field can be used to develop novel programming support tools that make use of psycho-physiological sensors. Depending on programming difficulties detected on the fly (*e.g.*, by identification of biometric patterns), appropriate interventions could be suggested by a rule engine.

Zusammenfassung

Ziel dieser Arbeit ist es, einen Visualisierungsprototypen zu entwickeln, der psycho-physiologische Daten aus Codeverständnisaufgaben geeignet darstellt. Verschiedene Studien der psychologischen Forschung haben gezeigt, dass psycho-physiologische (biometrische) Daten wie EEG, EDA oder Eye-tracking Daten Rückschlüsse auf kognitive Aktivitäten zulassen. Aufgrund dieser Tatsache können psycho-physiologische Sensoren dazu eingesetzt werden, um auf den Schwierigkeitsgrad von bestimmten Codeverständnisaufgaben zu schliessen.

Ein Datensatz, bestehend aus EEG, EDA und Eye-tracking Daten, wurde mithilfe der eigens zu diesem Zweck entwickelten Visualisierungen auf Datenmuster und Auffälligkeiten hin untersucht (z.B. zeitbezogene biometrische Muster oder Lesepfadmuster). Die gefundenen Resultate wurden anschliessend Signifikanztests unterzogen. Neben bereits bekannten Muster bzgl. des Code-Lesepfads wurden beispielsweise zeitbezogene Auffälligkeiten entdeckt. So gehen Anhäufungen von Lidschlägen oft einher mit relativ tiefen Messwerten für den mentalen Fokus (Attention eSense^{TM}-Werte). Zusätzlich wurden speziell die Messdaten von ähnlich konstruierten Aufgabenpaaren miteinander verglichen und die Resultate interpretiert. Dabei wurde u.a. festgestellt, dass beim Gebrauch von sogenannten mnemonischen Variablennamen signifikant weniger *"retrace declaration patterns"* (Blickbewegungen zur Variablendeklaration) zu beobachten sind.

Die Erkenntnisse darüber, wie Softwareentwickler Quellcode lesen und wie sie auf bestimmte Codeaufgaben reagieren, können dazu beitragen, problematische Codestellen zu identifizieren. Resultate aus diesem Forschungsbereich könnten dazu genutzt werden, neuartige Unterstützungstools zu entwickeln, die aufgrund der identifizierten Datenmuster Codeänderungen oder anderweitige Eingriffe vorschlagen könnten.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

1.1 Motivation

Performing software coding activities requires varying levels of cognitive effort depending on the overall difficulty of a given task or the complexity of specific code segments [PD10]. Various research studies (*e.g.*, [Doe57, Ax53, Tep02]) in psychology have shown that psycho-physiological measures as for example electroencephalography (EEG) or electrodermal activity (EDA) can be used to indicate cognitive states and efforts. In the field of human factors in software development, current research is exploring the use of psycho-physiological measures to support developers [Cur14].

This thesis is based on a noval approach by Fritz *et al.* that aims to classify code comprehension tasks into easy and difficult ones using various psycho-physiological measures. Fritz *et al.* explore the potential of EEG, EDA and eye-tracking data to predict the difficulty of code comprehension tasks. Actually, the approach investigates whether psycho-physiological data can be used to determine if a task is considered to be easy or difficult by an individual developer. Machine Learning experiments have shown that an accurate prediction whether a participant would find a specific task to be difficult or easy is possible. The same underlying dataset that is used by Fritz *et al.* is also used for this thesis [FBM+14].

Learning how software developers read code and how they react on specific parts of source code could help to locate problematic code segments. Findings in this research field could be used for example to develop novel programming support tools that make use of psycho-physiological sensors. Depending on patterns detected on the fly, appropriate interventions could be suggested by a rule engine. For instance, if based on the eye-movements an extraordinary high number of retrace movements to the variable declaration is recognized for a specific variable, renaming that variable could be suggested. A better variable name makes remembering what the variable was related to easier, which reduces the number of retrace actions that have to be performed by the developer.

1.2 Goal

The focus of this thesis is on finding patterns in a dataset that consists of EEG, EDA and eye-tracking data. The data are recorded in a lab experiment with 15 professional software developers solving 6-8 code comprehension tasks of varying difficulty level. The study was carried out by the University of Zurich in cooperation with Microsoft Research. The pattern analysis can be described as exploring the recorded data for interesting facts that apply for multiple participants. This includes various kinds of patterns as for example patterns that can be related to code-reading techniques (fixation path patterns), to biometric data that was measured at the beginning of a task or at the end of a task (time-related patterns), or to findings that apply to some specific groups of tasks only (task property patterns).

To provide a framework for the identification of patterns in the said data, visualization approaches have to be developed beforehand. A visualization prototype allow for the selection of an experimental task and display the code of that task in alignment with visualized psycho-physiological measurement data. Based on the eye-tracking data, the psycho-physiological measurements can be related to specific code segments. In the prototype, visual elements that allow to put the measured data in relation with the corresponding code segments should be integrated.

So, the goals of the thesis can be summarized as follows:

- *Develop a visualization prototype* that supports the analysis of biometric data recorded from developers working on code comprehension tasks. A set of different visualization approaches is required to identify various kind of patterns (*i.e.*, time-related patterns, fixation path patterns, code segment specific insights).

- *Analyze an existing dataset using the implemented visualization approaches* and evaluate whether the obtained insights are statistically significant or not. In particular, metric differences regarding task pairs that had been specifically designed to provide different difficulty levels are analyzed and interpreted.

1.3 Research Questions

Derived of the thesis goals, three research questions are formulated and listed in the following. The first research question focuses on the visualization approach, whereas the second is in connection with the process of finding patterns. Additionally, the third research question asks whether code segment specific data (measurement data that can be related to specific areas of interest) can be used as an indicator for difficulty.

(RQ1) *How can psycho-physiological measurements be visualized to support the analysis of patterns in the data that can be related to difficulties a software developer might have experienced while working on a specific parts of the code?*

(RQ2) *Which reoccurring patterns within code comprehension tasks can be identified in psycho-physiological data using visual inspection?*

(RQ3) *Given a pair of tasks that have been specifically designed to provide two different difficulty levels, can we explain the difference in difficulty by the AOI measurements?*

1.4 Structure of Thesis

This thesis is structured as follows: In a first step, background information (*Chapter 2*) about physio-physiological data is given. In particular, it focuses on the measurements that are received from the psycho-physiological sensors used in the code comprehension study. The related work section which is presented in *Chapter 3* is categorized into three areas: eye movement research concerning code reading tasks, use of various biometric measures in software development research and the use of biometric data in other research fields.

Then, the data gathering and handling is described in *Chapter 4*. This includes a description of the experimental study setup, lists the measured data as well as various metrics that are computed and shows data transformation steps that are used to prepare the relevant data for the different visualization approaches. Additionally, the database schema that is used for the visualization prototype is given in this chapter.

Before the implemented visualization approaches are presented in detail, the application architecture is explained and the design and interaction concept describing how the application can support the pattern finding process is given in *Chapter 5*. Then, the application as such is presented in eight different use case scenarios that make use of the three implemented visualization approaches (*Chapter 6*).

After the description of the main application features, the data analysis and pattern finding process are subject of *Chapter 7*. Basically, the data analysis part is divided into two parts: 1) Identification of patterns by visual inspection and 2) evaluating patterns using statistical tests. Several analysis approaches were applied to the visualized data leading to various kinds of patterns and data insights. Each of these approaches is presented in a separate chapter:

- *Chapter 7.2*: Fixation Path Patterns
- *Chapter 7.3*: Participant-specific Analysis
- *Chapter 7.4*: Time-related Analysis
- *Chapter 7.5*: Comparable Task Pair Analysis
- *Chapter 7.6*: Task-property Analysis
- *Chapter 7.7*: Metric Dependency Analysis

Finally, the findings are summarized, the stated researched questions are answered and aspects of future work are given in *Chapter 8*.

Chapter 2

Psycho-Physiological Background

In this chapter, background information is provided that describes psycho-physiological measures. First of all, an overview of the most commonly used psycho-physiological measures used in today's research is given. The measures that are used in the experimental study this thesis is based on, are then briefly presented. This includes also descriptions of analytical procedures to relate the data with cognitive states and processes.

2.1 Overview

In various research fields, psycho-physiological measures are popular to study the human's cognitive states. Psycho-physiological measures can be described as physical signals of the human body that are responses to psychological changes [DG11]. Using appropriate equipment, these signals can be tracked in real-time.

Table 2.1 provides a categorized overview of the most frequently used psycho-physiological measures based on Dirican *et al.* [DG11]. Measures used in this thesis are highlighted in bold.

Brain related	Eye related	Skin related	Heart related
• **Electroencephalography (EEG)** • Event Related Brain Potentials (ERP)	• Electroocuolography (EOG) • **Eye Movements (Fixations, Saccades)** • **Eye blinks**	• **Electrodermal Activity (EDA)** or also known as **Galvanic Skin Response (GSR)** • Electromyrogram (EMG)	• Heart rate variability (HRV) • Heart rate (HR) • Blood pressure (BP) • Respiration

Table 2.1: Overview of brain-, eye-, skin- and heart-related psycho-physiological measures [DG11].

2.2 Electrodermal Activity (EDA)

2.2.1 Defintion

The term electrodermal activity (EDA) was introduced by Johnson and Lubin in 1966 and includes all electrical phenomena in skin [Bou12]. It can be said that it measures changes in the skin's ability to conduct electricity over the time. It is also known as galvanic skin response (GSR) and is one of the most frequently used biosignal in psychophysiology [Bou12]. This has multiple reasons: First of all, electrodermal recording allows easy recognition of distinct electrodermal responses (EDR) that can be related to certain stimulus intensity [Ras73]. Using this data, implicit emotional states can be identified in an easy way. In addition to the ease of obtaining stimulation, accurate data recording is possible with fairly inexpensive recording devices [Bou12].

2.2.2 Tonic and Phasic Phenomena

Electrodermal activity can be divided into the tonic phenomena (Electrodermal level: EDL) and the phasic phenomenea (Electrodermal response: EDR) [Bou12]. Whereas the tonic signal component vary slowly over the time and changes occur typically in a period of 10 to 60 seconds, the phasic component is associated with short-term events and can be used to identify time intervals in that stimuli has been provided [FM11, Aff14]. A more detailed explanation is given below:

Tonic Signal (Skin Conductance Level). The tonic component of the EDA signal can be described as the level of skin conductance in absence of any specific external stimuli or discrete environmental events. Typically, the tonic skin conductance level is slowly varying over the time depending on factors like the persons' psychological state, dryness of the skin or hydration [Aff14]. In particular, stress or mental demanding activities can rise up the tonic skin conductance [Nak11].

Phasic Signal (Skin Conductance Response). Changes in the phasic component of the EDA can be associated with external stimuli or discrete environmental events as for instance sound or smell. In the phasic signal, such events are apparent as peaks (Skin Conductance Responses: SCR). To identify possible external stimuli, the latency time regarding conductance responses which amounts to 1-3 seconds must me taken into account [Aff14].

In summary, the tonic phenomena describes "the smooth underlying slowly-changing levels", whereas the phasic component typically show "rapidly changing peaks" [Aff14]. In contrast to the EEG signal, the EDA signal is of lower frequency and is based on physiological processes that are relatively slow-changing. In addition to that, the analysis of phasic changes in the phasic signal focuses mainly on irregularly occurring single events and not on pattern identification based on frequency or amplitude [Bou12].

2.2.3 Analytic Procedures

For the data analysis, the tonic and phasic component must be extracted from the recorded signal. Because of the relatively short time durations of the tasks used in the experimental study, the focus is set on analyzing the phasic component of the EDA signal. It has to be mentioned that there is not always a distinct relationship between a stimulus and a peak that occurs in the phasic signal. The skin conductance responses that are not event-related are called non-specific skin conductance responses (NS-SCRs). Fortunately, it is possible to separate the artifacts from the signal in most of the cases based on the characteristic course of the phasic changes [BWJR14].

Compared to other physiological signals as for example the heart rate changes, the electrodermal responses have a relatively long latency time. Normally, the latency time amounts between 1 and 2 seconds. Venables and Christie proposed a time window between 1 and 3 seconds for a SCR that is followed by a distinct stimulus [Chr81]. It was found that the latency varies with body temperature [Bou12]. In cases of skin cooling, it can increase up to 5 seconds [Bou12]. This means in fact that body temperature should also be considered if it is desired to further reduce the mentioned time window. For the analysis that is done in this thesis, the time window proposed by Venables and Christie is used to identify code segments that can be related to stimuli.

Based on literature, electrodermal activity can be related to cognitive processes or states such as arousal, attention, stress or anxiety [Bou12, DSF07]. Studies have shown that EDA measures can be used as an indicator for cognitive load level and task difficulty level [Bou12]. For instance, Wilson *et al.* has found significant differences in electrodermal activity regarding tasks that demand different mental workload [Wil02].

2.3 Electroencephalography (EEG)

2.3.1 Definition

The EEG measurement is commonly used in medicine and various research areas. It can be defined as medical imaging technique that measures the neural activity of the human brain [Tep02]. The human brain consists of billions of nerve cells that are called neurons [Wag75]. The electrical activity of these neurons produces current flows. The summation of currents result in voltage fluctuations on the scalp, which can be recorded using medical equipment. This recording is typically done using multiple electrodes (small metal discs) placed on the head's surface [Tep02].

2.3.2 Frequency Bands

The raw EEG signal can be divided into frequency bands that are alsow known as brainwave types. Brainwaves can bee categorized into the following basic groups: beta (>13 Hz), alpha (8-13 Hz), theta (4-8 Hz), delta (0.5-4 Hz). Each of these brainwave types can be associated to different mental states [Tep02]. Since NeuroSky's proprietary metrics have been used for representing the EEG data in the visualization approaches, the mental states that can be associated with the brainwave types are not mentioned in this section.

2.3.3 NeuroSky's Attention and Meditation eSense™ Metrics

In the experimental study, the lightweight and easy to setup NeuroSky MindBand that is placed on the participant's forehead is used as an EEG measuring device. It must be said that the device uses a single electrode and therefore provides a coarser information in comparison to a EEG measuring technique where multiple electrodes are used [RMDMM+09]. But due to the fact that the device is very easy to setup, barely noticeable and therefore reduces some of the difficulties to collect accurate data, the NeuroSky MindBand is a very suitable device for the code comprehension study. Along with the raw EEG Signal, NeuroSky provides two separate measures named as *Attention* and *Meditation* which provide information about emotional and cognitive states [Neu09]. These values are computed using dynamically learning algorithms that are developed by NeuroSky eSense™ [RMDMM+09]. Both values are reported on a scale of 1 to 100 and are operating at 1Hz. Implications for values in certain value ranges are given in Table 2.2. The color of the cells in the last two columns indicate how values within certain ranges can be related to task difficulty (red: high difficulty, green: low difficulty). Below, the eSense™ metrics are described:

Attention. The *eSense™ Attention meter value* measures the level of mental focus. It can be said that the measure shows how concentrated somebody is at a given moment in time. Intense concentration and stable mental activity leads to high attention values whereas wandering thoughts or anxiety can reduce the attention value [Neu14].

Meditation. The *eSense™ Meditation meter value* represents the person's mental calmness. It must be said that it measures the relaxation on mental levels and not on physical levels. This means, that relaxing the muscles does not necessarily lead to high meditation values. High meditation values can be related to reduced active mental processes. In addition to wandering thoughts or anxiety, agitation and sensory stimuli can lead to lower values [Neu14].

eSense™ Value	Status	Attention	Meditation
1 - 20	strongly lowered	strongly distracted	strongly agitated
20 - 40	reduced	distracted	agitated
40 - 60	neutral	-	-
60 - 80	slightly elevated	mental focused	mental relaxed
80 - 100	strongly elevated	heightened attention	heightened calmness

Table 2.2: NeuroSky's eSense™ value ranges with corresponding implications [Dug13].

2.3.4 Eyeblinks

Although the EEG is designed to record brain activity, it is also possible to detect motor signal of the face because of the placement of electrodes on the scalp [AÇÇ09]. It was found that eyeblinks are one of the major sources of inferences in the EEG exam [SAdA08]. Fortunately, such artifacts can be distinguished from the neuronal activities [FBM+14]. In neuroscience research, it was found that an increased blinking rate can be used as an indicator for arousal [DJM90].

2.4 Eye-tracking

2.4.1 Tracking Methods

There exists several techniques to record eye-movements. Basically, they can be categorized as follows: 1) recording eye rotations using so called in-eye sensors, 2) eye tracking based on video recording and 3) measuring of electric potentials using electrodes placed near the eyes [Tor07]. Actually, the technique that is listed second was used in the code comprehension study. In this method, a camera focuses on the eyes that are illuminated with light (typically infrared). Using the camera, the reflection of the light can then be recorded. Based on these reflections as well as the center of the pupil, eye-movements can be determined [Tor07].

2.4.2 Gaze Event Types

Eye movements are typically analyzed in terms of gaze event types that are described below:

Fixations. A fixation is recognized when a person focuses on a specific area of interest and pauses in that informative region [SG00]. A common metric that is related to fixation is the fixation duration which is also provided by the eye-tracker that was used in the experimental study this thesis is based on.

Saccades. The rapid movements between the fixations are called saccades [SG00]. For these saccades, no specific area of interest can be assigned. In literature, the terms "saccadic jumps" or "saccadic eye movements" are often used for this fast change of the eyes' position [FWL59]. The number of saccades over a specific period of time is a popular measure that is used as an indicator for various phenomena. Goldberg *et al.* has shown in a user interface evaluation study that a relatively high number of saccades indicates that the interface tends to be poor [GK99].

2.4.3 Pupillometry

Besides the fixation durations and the number of saccades, the pupil size is an important measurement that allows conclusions about cognitive states and processes. In psychology, the term pupillometry is used for the measurement of the pupil diameter [KKH08]. Studies have shown, that especially the peak amplitude of the pupil size can be used as an indicator for memory load [Bea82, Kli10]. Note that for analyzing pupil diameter data, only the data of the dominant eye of the subject should be considered [CH06]. This is because of the so called ocular dominance that describes "the tendency to prefer visual input from one eye to the other" [KC01].

Chapter 3

Related Work

The related work section is classified into three areas: First of all, research concerning code reading techniques and eye movement patterns is given. After that, general findings on the use of psycho-physiological metrics in various research fields are listed, before it is then focused on software development research using psycho-phyisological measures.

3.1 Code Reading Patterns

There exist a number of studies where eye-tracking is used to analyze how code reviewers read software code. Uwano *et al.* have proposed an objective way to characterize the performance of participants in reviewing software code by using eye movement data. This includes an integrated measuring environment for line-by-line tracking of eye movements [UNMM06]. As a result of this study, three patterns were found: The scan pattern, the retrace declaration pattern and the retrace reference pattern. These patterns are briefly exlpained below.

Scan Pattern. It was found that most of the participants read the code from the top to the bottom briefly before concentrating on some specific parts of the code. The statistics in Uwano's study shows that in the first 30 percent of the code review time, nearly 73 percent of the total lines of code were watched in average. Uwano names this preliminary reading as *scan pattern* [UNMM06].

Retrace Declaration Pattern. In addition to the scan pattern, Uwano *et al.* found that when a participant reaches a line of code where a variable is initially used, the subject often focuses the line where that variable is declared within a short time period. This pattern, that is named as *retrace declaration pattern*, is interpreted as a cognitive action where the participant wants to be sure about the data type [UNMM06].

Retrace Reference Pattern. The *retrace declaration pattern* has similar characteristics as the retrace declaration pattern. It describes the eye movement from a specific line of code where a variable is defined back to the line where the variable is referred. Uwano argues that this behaviour is shown to remember or to recalculate the value of a specific variable [UNMM06].

Illustrated examples of the patterns that are described above can be found in the data analysis part of this thesis (see Chapter 7.2). In addition to these patterns, Uwano *et al.* found in a quantitative analysis that participant with a relatively short scan time tend to take more time for detecting errors in the source code. That means, that the longer a participant spends in the initial scan, the quicker the error is found [SFM12]. In the mentioned study, the first scan time is defined as the time spent from the beginning of the task until 80 percent of all the lines are read [UNMM06]. Another study by Sharif *et al.* that follows a similar experimental setup but with a higher number of participants makes the conclusion that scanning time significantly inversely correlates with error detection time and the total number of fixations. In addition to that, Sharif *et al.* found in a replication study that programming novices tend to read lines more broadly than programming experts. Also the amount of time to scan the code is higher for the novices. The found differences are explained by the experts intuitive notion of detecting problem areas [SFM12].

3.2 Psycho-physiological Research

There exist several studies that investigate the use of various psycho-physiological measures to characterize a person's mental or emotional state and processes. Table 3.1 provides an overview of cognitive processes that can be related to different psycho-physiological measurements [FBM+14].

Measure	Metric	Findings related to Cognitive States and Efforts
EEG	Attention	Mental focus, concentration, cognitive load [RMDMM+09]
	Meditation	Mental calmness, relaxation [Neu14]
	Frequency bands	Can be connected to different mental states [Tep02]
	Eyeblinks	Inverse correlation with visual attention or mental load [DJM90], indicator for stress or anxiety [Doe57, HBPR09], arousal corresponds to increased eyeblink rate [DJM90]
EDA	Tonic signal	Mental workload [Wil02], anger & fear [Ax53]
	Phasic signal	Anger & fear [Ax53], arousal [MKK+12]
Eye-tracking	pupil diameter	Diameter of peak amplitude indicates memory load, pupillary response varies with task difficulty [Bea82], pupil dilation as a measure for workload [BI08]
	Saccades	Regarding user interfaces: A relatively high number of saccades indicates a poor interface [GK99]
	Fixations	Fixation duration used as a indicator for the difficulty of extracting information from a display [JK03]

Table 3.1: Psycho-physiological measures and their use as indicators for various mental states.

3.3 The Use of Psycho-physiological Measures in Software Development Research

Apart from the already presented related research concerning code reading techniques using eye-tracking systems, there exist a few other studies in the field of software development that make use of psycho-physiological measures. In the following, a selection of studies that investigate software engineering aspects is briefly presented.

Identifier-naming Conventions. Sharif *et al.* have investigated the effect of using identifier-naming conventions "camel-case" and "underscore" on code comprehension with the help of eye-tracking. Results have shown that there is no difference between the two conventions in accurancy. However, a difference concerning recognition time and visual effort is found: Using identifiers in the underscore style, participants tend to recognize them faster and lower visual effort is required [SM10].

Experienced-based Differences regarding Code Reading. Crosby *et al.* have examined in a code comprehension study how participants view short algorithms. In particular, the differences regarding code reading techniques between low- and high experienced developers are discovered using eye-tracking. It was found that less experienced software developers took more time on comments but less time on complex statements than advanced developers [CS90].

Influence of Programmers' Mood. Khan *et al.* have investigated how developers' mood affect their programming activities. Although no psycho-physiological sensors were used in this study, it examines psycho-physiological aspects. In lab experiments, the developers' mood was manipulated using low-arousal-evoking and high-arousal-evoking video clips a mood change validation tests was performed. Results for debugging tasks have suggested that developers' performance were influenced by their mood [KBH11].

User Interface Evaluation based on Eye-tracking. In a user interface evaluation study, Goldberg *et al.* uses eye movement locations and scan paths to evaluate the user interface of a drawing tool. Good and poor interfaces are created *i.e.*, by various grouping of buttons and icons. Results have shown that well-organized functional grouping lead to significantly shorter scan paths and a lower number of fixations [GK99].

Detection of sub-vocal Utterances. Parnin *et al.* investigates the use of electromyopgrahy (EMG) for software developers. Using EMG, occurrences of so called sub-vocal uterrances can be detected. The phenomena of sub-vocal uterrances can be described as electrical signals that are sent to vocal cords or the tongue. Parnin *et al.* have found that this measure might be used to determine the complexity of a task [Par11].

Most of the research studies that are described above are limited to eye-related psycho-physiological measures. Little research has investigated the use of other measures as for example electroencephalography (EEG) or electrodermal activity (EDA) in software development. However, Züger has shown in her thesis that EEG data can be used to determine between two levels of difficulty for code comprehension tasks. In addition to that she suggested use cases in relation to how brainwave data can be used to support software developers. Concrete use cases with regard to self-monitoring or the integration of a reward system are described in her work [Zü13].

Fritz *et al.* goes further and investigates also the use of eye-tracking and EDA data to classify code comprehension tasks based on the difficulty. In the approach this thesis is based on, it was shown that psycho-physiological measures such as eye-tracking, EEG or EDA can be used to determine whether a task is considered to be easy or difficult. Machine Learning experiments have shown that the task difficulty (easy or difficult) for a new participant can be predicted with a precision of 65% using eye-tracking-, EDA- and EEG data [FBM+14]. For predicting a new task the precision amounts to nearly 85%. It was shown that also using fewer sensors, an accurate predication whether a task is perceived as easy or difficult is possible (*e.g.*, prediction by participant, precision using eye-tracking data only: 69.16%) [FBM+14].

Chapter 4

Data Processing and Storing

In this chapter, relevant data processing steps to integrate the existing dataset into the visualization prototype are explained. First of all, a description of the experimental study setup is given and the used psycho-physiological measures are listed. In addition to that, it is briefly described how additional metrics are computed from the raw data. Finally, an overview of the database schema that is used by the visualization prototype is given.

4.1 Experimental Study Setup

4.1.1 Subjects and Measuring Devices

The underlying dataset used in this thesis was obtained in a lab experiment[1] conducted by Microsoft Research in cooperation with the Unveristy of Zurich. In the study, 15 participants who have at least two years of programming experience solved 6-8 short code comprehension tasks of different difficulty level. In total, data for 116 tasks was obtained. Below, the measuring devices used in the experimental study are listed:

- *Eye-tracker* to record fixation/saccade and pupil diameter information
- *NeuroSky MindBand device* placed on the forehead to record EEG. Additionally, attention and meditation level data are provided, computed based on the brainwave data.
- *Wrist band* (on the non-mouse holding hand) to measure EDA and body temperature

A more detailed overview of the measuring devices along with the corresponding measurements as well as further metrics that can be computed out of this data can be found in Table 4.1, Chapter 4.2.

[1] For the replication package of the experiment, see: http://research.microsoft.com/apps/pubs/?id=209878

4.1.2 Experimental Tasks

All the code comprehension tasks used in the study are written in C# and are basically about drawing shapes on the screen. The tasks were constructed to have a varying level of difficulty. Based on the kind of task, it can be classified in two task types. These types are briefly described below and illustrated in Figure 4.1.

- **Task Type A:** In task A1 and A2 two rectangular objects are drawn on the screen based on specific coordinates of the rectangles' corners. The participant was asked to figure out whether these rectangles overlap or not. Task A2 is constructed in a more complex way than task A1. So it is expected that the participants experience more difficulties in task A2 than in task A1.
- **Task Type B:** In tasks B1, B2, B3, B6, B9 and B11, various shapes as for example circles or rectangles are drawn. For each of these code comprehension tasks, the participant was asked for the correct order in which the objects were drawn on the screen. The tasks varied for example in the order of variable assignments, the mathematical complexity, kind of variable name or the complexity of loop statements. More details can be found in Chapter 7.5.

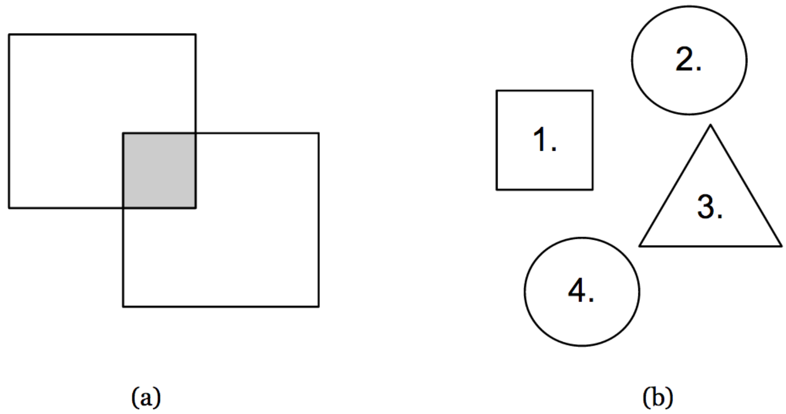


Figure 4.1: Illustration of task types: tasks of (a) type A are about overlapping rectangles and in tasks of (b) type B it is asked for the order of the drawn shape objects.

The fact that the tasks with the same task type are quite similar but differ in their difficulty level allows a pairwise comparison between some of them. These comparable tasks are named in this thesis as task pairs. A detailed list along with difference descriptions can be found in Chapter 7.5.

For each task, a duration of 2 to 5 minutes was planned, but results have shown that some of the tasks were solved in less than 30 seconds (*i.e.*, task B1). The longest average task duration was noticed for task B6 (3.85 min). In total, 67.2 percent of the tasks were solved correctly. In order to reduce inferences, a fishtank movie is shown for 120 seconds before each of the tasks, so that the gathered data is influenced by the task he solved before as less as possible. To mitigate the learning effect, the task order is randomized for each participant.

4.1.3 Perceived Difficulty of Tasks

After each task in the experimental study, a written NASA TLX survey is conducted [HS88]. A NASA TLX survey includes a set of workload-related human factors that have to be rated by the participant. The subscale ratings are combined and a perceived difficulty value is computed [HS88]. In the end of the experiment, the participant was asked to rank all the tasks based on the perceived difficulty. To get an overview of the difficulty of the individual tasks, the resulted rankings are illustrated in Figure 4.2. The dots that are colored in red represent tasks that were not solved correctly, whereas black dots indicate that the task is completed successfully. Mean ranking values are shown as vertical lines in each lane of a corresponding task. Based on Figure 4.2 following statements regarding the perceived difficulty of the tasks can be made:

- Task B1 is ranked as the easiest task, followed by task B2.
- 7 Participants ranked task B6 as the most difficult task.
- Although no participant completed task B9 successfully, two participants ranked it as the second easiest task (implies that perceived difficulty $\neq$ difficulty).
- Task A1 has the highest deviation in perceived difficulty.
- Order of tasks based on mean perceived difficulty: B6, A2, B11, B9, A1, B3, B2, B1.

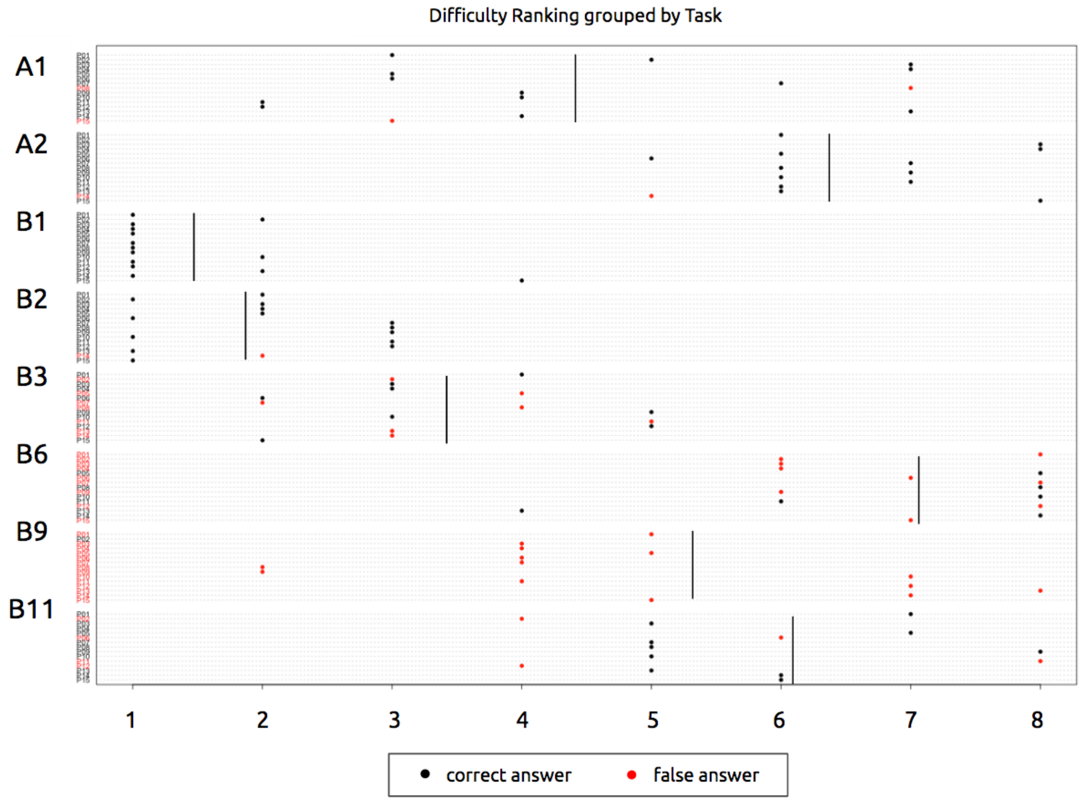


Figure 4.2: Dot plot representing perceived difficulty data for each participant.

4.1.4 Areas of Interest (AOIs)

For all the tasks so called *areas of interest* are defined. These can be single lines of code or code blocks. If the eye-tracker recognizes that the subjects' focus is in a specific area of interest, the flag for that segment is set to true for this data entry. Figure 4.3 illustrated two code examples where the defined areas of interest are highlighted using colored rectangles.

Figure 4.3: Example tasks with AOIs that are highlighted with colored rectangles: AOIs for task B1 on the left and AOIs for task B2 on the right.

The defintion of these areas of interest allows code segment specific comparison. For instance, the object initialization in task B1 can be directly compared with the object initialization part in the comparable task B2. This AOI-specific comparison allows to discover psycho-physiological insights for varying code segment difficulty.

4.2 Measured Data and Computed Metrics

4.2.1 Overview

Table 4.1 gives an overview of all the psycho-physiological data that is measured in the experimental study. In the last column, metrics that are computed using Matlab scripts are listed. A description how these computations are performed is given in the section that follows the table below.

Measuring Device		Measured Data	Computed Metrics
	**Measure:** EEG **Device:** NeuroSky's eSense^{TM} **Sampling Rate:** 512Hz	Attention eSense^{TM}, Meditation eSense^{TM} Raw EEG signal	Brainwave frequency bands, Eyeblinks
	**Measure:** EDA **Device:** Affectiva Q Sensor 2.0 **Sampling Rate:** 8Hz	EDA signal Body temperature, Acceleration data	Tonic and phasic signal component, phasic EDA peaks
	**Measure:** Eye-tracking **Device:** Tobii TX300 EyeTracker **Sampling Rate:** 300Hz	Pupil size left, Pupil size right Gaze event type, Gaze event duration	Pupil diameter peaks are computed for the dominant eye

Table 4.1: Overview of measuring devices, recorded data as well as computed metrics [Min13, ia13, Tec13].

4.2.2 Metric Computations

The computations steps that are performed to get the metrics listed in the last column of Table 4.1 are briefly described below:

Brainwave Frequencies and Eyeblinks. To retrieve the brainwave frequency bands, the EEG data is split using a Fast Fourier Transformation (FFT). Also the eyeblinks can be detected based on the EEG data. This is the case due to the placement of the EEG device: Motor activities of the face as for example are eyeblinking are identified as artifacts that can be distinguished from the neuronal activities [FBM$^+$14]. So, in addition to the brainwave frequency bands, the eyeblinks are also computed based on the EEG data. For that, a band-pass Butterworth filter (0.5 Hz - 3Hz) is used [FBM$^+$14]. Then, an algorithm that detects peaks [2] in a given signal is applied. Peaks that are more than 100 times stronger than the mean amplitude of the waveform are then considered as eyeblinks.

Tonic and Phasic Component of EDA Signal. The EDA signal consists of two parts: The low frequency tonic component and the high-frequency phasic component. After removing noise from the raw EDA-Signal using exponential smoothing filtering, butterworth filters are applied to extract the tonic and the phasic component of the signal. For the tonic signal a low-pass Butterworth filter (5th order, 0.05Hz) is used, whereas for the phasic component, a high-pass Butterworth filter is applied. Because the phasic signal reflects reactions based on exteranl stimuli, the peaks of the phasic component can be defined as an important measure [FBM$^+$14]. This meas that detected peaks can be associated with short-term psycho-physiological events. Like for the eye blink detection, the Matlab-based peak finding algorithm is used to detect peaks in the phasic component of the EDA signal (min. amplitude: 0.02 μS [Bou12]).

Pupil Diameter Peaks. Based on the pupil diameter data of the participants' dominant eye, data points with extraordinary high pupil dilation are extracted from the dataset. Again, the Matlab-based peak finding algorithm is used for that. Before applying the algorithm on the dataset, some preprocessing steps are required: First of all, data points that are tagged as invalid by the eye-tracker are removed. Then, the first data entry for each fixation that is recorded after an eyeblink has to be eliminated. Removing this entry is required to not distort the data since the pupils dilate a little bit when the eyes are closed (even if this is only for a short time) [FBM$^+$14]. Finally, three types of peaks that are categorized based on the peak size are determined: Peaks that correspond to a pupil diameter that is 0.4 mm, 0.2 mm and 0.1 mm above the mean pupil diameter.

[2] PeakFinder: http://www.mathworks.ch/matlabcentral/fileexchange/25500-peakfinder/content/peakfinder.m

4.3 Dataset Transformation

4.3.1 Overview of Relevant Visualization Data

For the various visualization approaches, the measured data has to be prepared in an appropriate way. For that, data extraction and transformation steps are required. Basically, the visualization data can be categorized into three data categories. These categories along with examples what kind of data that can be associated with each category is given below:

- **AOI Measurement Data:** For each participant, the metric values for all the AOIs that are involved in one of the tasks a participant worked on, are computed. This is done using time- and memory intensive Matlab calculations. For example, the mean attention value of a specific code segment is calculated by all the attention values that have been measured when a hit in that specific code segment was recognized. The resulted metric values are finally used for the *Grid View*; see Chapter 6.4. A list of all defined AOI metrics can be found in Appendix C. In the following ,an extact of a sample JSON document (returned from a web service request) that contains measurement data for the AOI named `A1_AOILine8Hit` is given.

```
{
    "value": 172830.9914,
    "metric": "BetaOverAlphaPlusThetaDiffTFTWithinAOI",
    "metricID": 41,
    "aoiName": "A1_AOILine8Hit",
    "participant": "P06"
},
{
    "value": 3.229,
    "metric": "FixationsPerMinuteWithinAOI",
    "metricID": 60,
    "aoiName": "A1_AOILine8Hit",
    "participant": "P06"
},
```

Listing 4.1: Extract of sample JSON document representing AOI metrics.

- **Timeline Data:** For the visualization, it is suggested to implement a timeline approach that displays how the various sensor data vary over the time for a specific task and participant. Data that is displayed on the timeline includes *e.g.*, fixation data, eyeblinks, MindBand values, EDA values, pupil diameter data a.s.m. For more details about the *Timeline View*, see Chapter 6.2. The listing below shows an extract of a sample JSON document that contains timeline data.

```
{
    "timestamp": "2013-01-21T09:45:58.000Z",
    "attention": 30,
    "meditation": 34
}
```

Listing 4.2: Extract of sample JSON document representing timeline information.

- **Task Information Data:** Along with the visualization, a information container that is filled with statistical information as for example the perceived difficulty or the information whether the task is solved correctly or not is given. A sample JSON document that contains information data for task B1, solved by participant P02, is given in the listing below:


```
{
    "A_TaskName": "B1",
    "A_ParticipantName": "P02",
    "Position_in_Task_Order": 1,
    "Is_Correct_Answer": [1],
    "avg_Meditation": 57.878,
    "min_Meditation": 27,
    "max_Meditation": 88,
    "avg_Attention": 29.6321,
                [...]
    "Task_Completion_Time": 33,
    "Task_Difficulty_Ranking": 2,
    "NASA_TLX": 6.3299999237,
    "NASA_TLX_Rating_Performance": 12,
    "NASA_TLX_Rating_MentalDemand": 5,
    "NASA_TLX_Rating_PhysicalDemand": 2,
    "NASA_TLX_Rating_TemporalDemand": 5,
    "NASA_TLX_Rating_Effort": 3,
    "NASA_TLX_Rating_Frustration": 3
}
```


Listing 4.3: Extract of sample JSON document representing task information.

4.3.2 Metric Computations by Areas of Interest

AOI specific measurements are used for the *Grid View*. The *Grid View* presents various metrics for specific code segments in a table-like form. These AOI metrics are defined based on the metrics that are used by Fritz *et al.* (computation of metrics per task). Table 4.2 lists some of the AOI metrics defined in this thesis along with a short explanation of how they are calculated. A complete list can be found in Appendix C. For the value calculations, Matlab scripts are created that automatically fetch table entries for a specific area of interest and participant. Using AOI specific data, as well as corresponding baseline data, values for all the defined metrics are computed.

Measure	AOI Metric Name	Description and Computation
EDA	Δ Mean tonic	(Mean tonic value during mind relaxation phase) - (Mean tonic value of data entries where participant's fixation is in AOI)
	Δ Mean phasic peak amplitude	(Mean phasic peak amplitude during mind relaxation phase) - (Mean phasic peak amplitude for peaks that occurred while participant focused on AOI)
EEG	Δ Mean Attention	(Mean Attention during mind relaxation phase) - (Mean Attention while participant focused on AOI)
	Δ Mean Meditation	(Mean Meditation during mind relaxation phase) - (Mean Meditation while participant focused on AOI)
	Δ Eyeblinks per second	(Eyeblinks per second during mind relaxation phase) - (Eyeblinks per second while participant focused on AOI)
Eye	Mean fixation duration	(Sum of fixation durations for specific AOI) / (No. of fixations for specific AOI)
	Mean pupil size	Mean pupil size of dominant eye while participant focused on specific AOI
	[...]	[...]

Table 4.2: Selection of AOI metrics (7 of 45) with computation descriptions

To understand how this metric computation is implemented in Matlab, the computation for a sample metric (Δ*Number of phasic peaks per second*) is shown in the following step-by-step. The computed values for this AOI metric represent the number of peaks that can be related with fixations in the area of interest under consideration, normalised by total fixation duration of the AOI and the baseline data.

1. Import data from the relational database using Matlab database Toolbox[3]: The relevant data entries of a specific task and the corresponding fish tank phase (mind relaxation phase) are retrieved from the MySQL database.
2. Then, the EDA signal is cleaned. This includes a DC Shift correction to base the signal at 0 micro Siemens. Additionally, noise is removed by applying an exponential smoothing [FBM$^+$14].
3. In a further step, the EDA signal recorded during the fish tank phase as well as the EDA signal that is recorded while the participant was working on the code comprehension phase is extracted from the imported data. Butterworth filters are applied to split these EDA signals into its phasic and tonic parts. The phasic signal of the task is then stored in a variable named as `phasic_task`.
4. Then, for each of the AOIs that are part of a given task, the metric values are computed one after the other.
   (a) First, the data entries that are tracked while the participant focused on the AOI under consideration is extracted from the retrieved task data. The timestamps of these extracted entries are stored in a separate array named as `aoi_timestamps`.
   (b) For the metric under consideration (Δ*Number of phasic peaks per second*), the peaks of the phasic signal that can be related to a given AOI have to be determined. This is done using the function shown in Listing 4.4.

```
function peakContainer = getAOIPeakIDs(eda_task, phasic_task, aoi_timestamps)

     % Retrieve timestamps of peaks in the phasic signal
     peak_timestamps = getPeakTimestamps(eda_task, phasic_task);

     % Iterate over all the peaks & check whether there is any
     % AOI timestamp in the stimuli time range of the peak.
     % Stimuli time range of a peak:
     % Start: Timestamp of peak - 3 s
     % End: Timestamp of peak - 1 s

     peakContainer = [];

     for i = 1:size(peak_timestamps)
        [peakStart,peakEnd]= getPeakRange(peak_timestamps(i),3,1);
        isPeakInAOI = checkPeakDatetime(peakStart, peakEnd, aoi_timestamps);
        if isPeakInAOI == true
          peakContainer(end+1)= i;
        end
     end
end
```

Listing 4.4: Extract of M-File to determine whether found peaks can be related to specific AOI.

[3]http://www.mathworks.ch/products/database/

To figure out whether the external stimuli (can be associated with phasic signal peak) occurred while the participant focused on a specific AOI, it must be checked whether the participant's focus during the time interval in which the stimuli can be located was on the AOI or not. This means that it is checked for each peak whether there is any AOI timestamp in the time window range of a probable stimuli (start: timestamp of peak - 3 s, end: timestamp of peak - 1 s). The resulted peak information is then returned.

(c) Finally, the number of peaks that can be associated with the specified AOI is normalised over the total time the participant focuses on that AOI. The Δ-value is then computed by subtracting the normalised AOI-specific value from the time normalized number of peaks in the fish tank phase. The required steps to normalize the number of phasic EDA peaks by the fixation durations and the baseline data respectively, are shown in Listing 4.5.

```
numberOfAOIPeaks = size(peakContainer,1);

     aoiDuration = size(aoi_timestamps,1)*0.125;

     if (numberOfAOIPeaks ~=0)

         nPeaksFT = calculateNoEDAPeaks(phasic_ft);

         if (nPeaksFT ~= 0)
            fFTValue = nPeaksFT/60;
         else
            fFTValue = 0;
         end

         fAOIValue = numberOfAOIPeaks/aoiDuration;
         fValue = fFTValue - fAOIValue;
```

Listing 4.5: Extract of M-File that normalizes the number of phasic peaks.

The computation process of an AOI metric value described above is performed for all the participants over all the tasks and all the AOIs that are defined within a task. Using loops, each of the 45 metric computation process is started automatically (see Matlab script `FeatureGenLoop.m`). How the computed AOI measurements are used in the visualization prototype to discover AOI specific insights is presented in Chapter 6.4.

4.3.3 Database Schema

Figure 4.4 gives an overview of the required data processing steps and shows the data tables that are created for the visualization prototype. There exists two main approaches how the data entries are generated. On the one hand, Matlab scripts are run to get the computed metric values. This includes for example the calculation of the AOI metrics as well as the computation of eyeblinks, pupil diameter peaks or peaks in the phasic component of the EDA signal. This data processing step is highlighted in Figure 4.4 using a grey rectangle. Because of performance reasons, data requests are not performed on the raw table. Instead of that, basic SQL scripts are created that extract data from the raw table and create smaller tables containing only relevant visualization data. Latter approach is performed for all data where no additional computations are required (*e.g.*, attention/mediation data, fixation data and task information data).

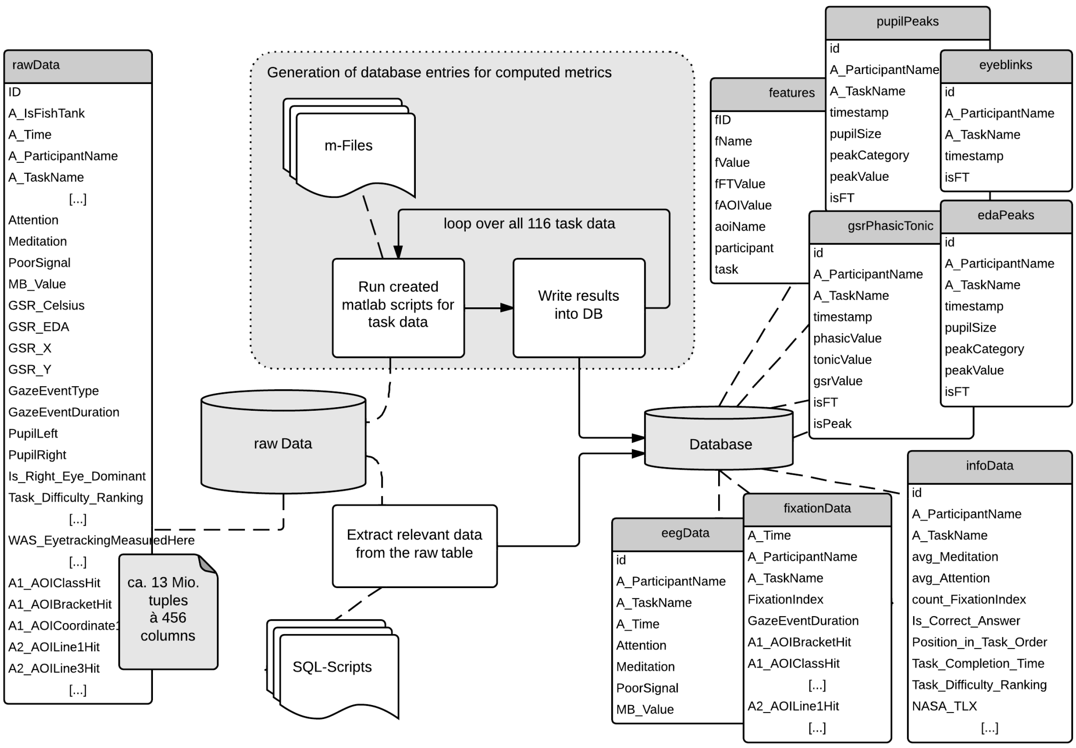


Figure 4.4: Overview of generated tables for visualization prototype and how database entries are generated using computation and extraction processes.

Chapter 5

Visualization Prototype

This section presents mockups of the protoype as well as the architecture of the implemented application. In addition to that, related visualization approaches are given. The visualization tool as such is then presented in form of use case scenarios in a separate chapter.

5.1 Related Visualization Approaches

5.1.1 Fixation Path Visualizations

There exists a number of studies on eye-tracking, in which eye movements on a source code is the center of investigation. Some of the visualization approaches that were used to gain data insights are presented in this chapter. Uwano *et al.* developed a software application named *Crescent* (Code Review Evaluation System by Capturing Eye movemeNT), which allows the visualization of the line-by-line fixation data using the result viewer [MUOM09]. A sample result viewer screen is depicted in Figure 5.1. In this visualization, the X-axis captures the time spent on a code review task. The grey bars mark fixations in the corresponding lines of a code. The width of a bar indicates the duration of a fixation.

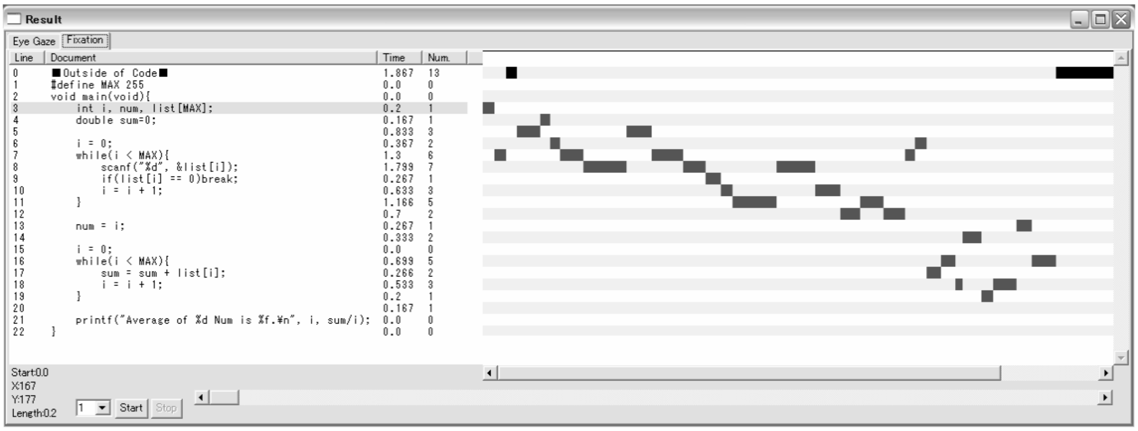


Figure 5.1: Line-by-line fixation visualization in the result viewer of the *Crescent* application [MUOM09].

To identify patterns that focus on the fixation path and are independent of fixations duration, Uwano *et al.* proposes a visualization that is illustrated in Figure 5.2. In this case, the X-Axis represents the fixation index. The fixation durations are omitted.

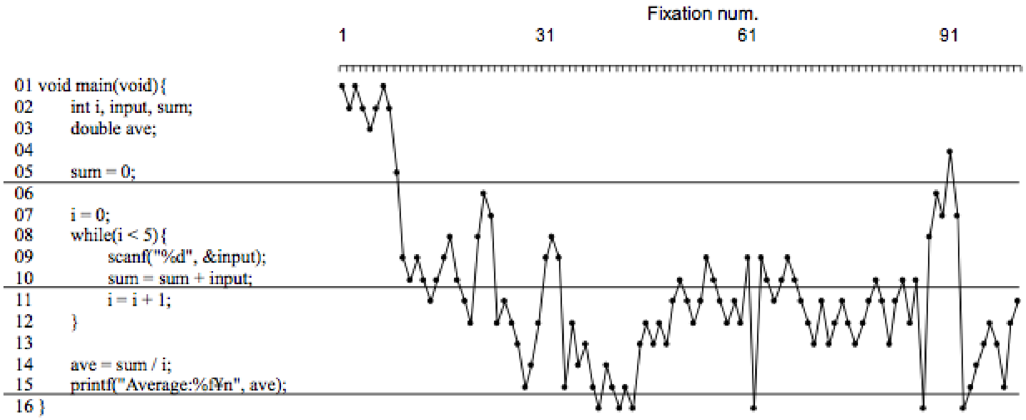


Figure 5.2: Line-by-line fixation visualization with focus on fixation path [MUOM09].

An eye-tracking enabled software traceability environment called *iTrace* that is being developed by the Software Engineering Research Lab of the Wichita State University allows to record and visualize eye-tracking data for various purposes. Figure 5.3 shows in two examples how the iTrace environment can be used for eye-tracking data on source code [WSU14]. Both approaches are well suited to identify those areas of the code with a high density of fixations. The heat map approach (depicted on the right-hand side of Figure 5.3) is an ideal approach for fixation density analysis. However, the approach bears the disadvantage that path information is not preserved in the visualization. For the visualization prototypes of this thesis, fixation visualizations as shown in Figure 5.3 present no practicable scheme, because the dataset at hand contains only line-by-line tracking information, rather than accurate eye location information.

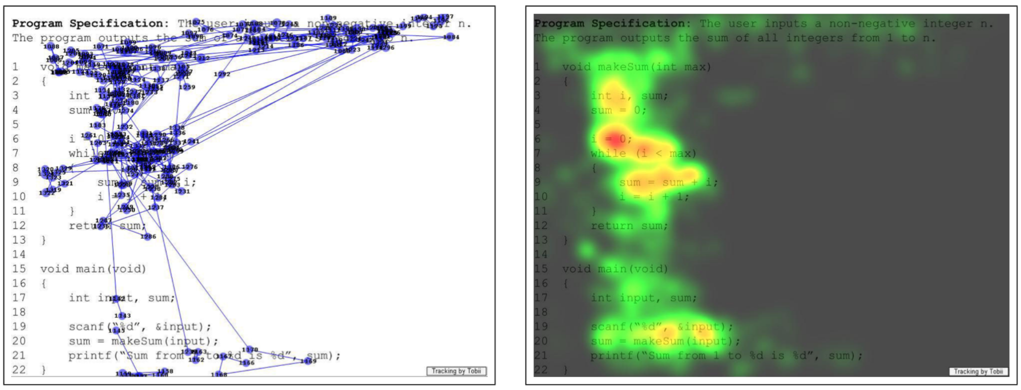


Figure 5.3: Eye-tracking on source code visualizations generated using the *iTrace* environment [WSU14].

5.1.2 EEG, EDA and Pupillometry-related Visualizations

In research, EEG, EDA, and Pupil Diameter data is often visualized as simple line charts [CSPM10, JN11]. Software applications specialized in biometric data provide various features to support data analysis. *Biopac's AcqKnowdledge software*, for example, is an EDA analysis software and includes options to locate skin conductance responses (SCR). Figure 5.4 [BWJR14] provides a sample screenshot with highlighted skin conductance responses (SCR).

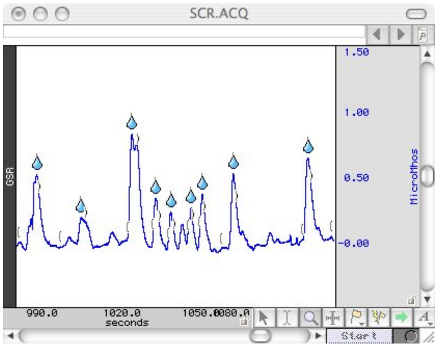


Figure 5.4: EDA data visualized in *Biopac's AcqKnowdledge software* (drop icons represent SCRs) [Ins14].

5.2 Design and Interaction Concept

The main requirement for the prototype can be summarized as follows: Visually display the code that was used in specific code comprehension tasks and in alignment to that code, various psycho-physiological measurements that were recorded during the code comprehension task. Additionally, the visualization should allow to relate measurement values to specific parts of the code based on the eye tracking data.

In an early phase of this project, several UI sketches and mockups were conceived and contemplated in order to show how visualizations can help to analyze the present data. A selection of these mockups is given in this chapter. Already existing data visualization approaches handling similar data were taken into account (*e.g.*, eye-tracking data recorded for software developers during code reviews). In each elaborated visualization approach, the code that was used in a specific code comprehension task is shown. In alignment to that code, the corresponding psycho-physiological data should be visualized in an appropriate way. In a first phase, two main visualization approaches are found to be suitable to support the data analysis:

- A first visualization approach called *Grid View* shows AOI specific metric data over multiple participants. This means that all measurement data that is recorded while specified participants focused a specific AOI is collected for the computation of aggregated metric values. This approach can be used to explore AOI-specific insights.
- Another approach named as *Timeline View* is used to display data over task duration for a specific task and a specific participant. This allows to discover task- and participant specific insights.

In the following sections, the concept behind the data visualization ideas is described and illustrated with corresponding mockups.

Grid View. The *Grid View* consists of a table-like structure and shows various metrics for specific areas of interest (AOIs). A column represents a specific metric (*e.g.*, Δ*Mean Attention*). Each data entry with a timestamp at which the participant has focused on a specific code segment is taken into account for the computation of the metric value for that specific area of interest. Based on the computated value, the corresponding cell is colored accordingly. The coloring scheme ranges from red for values that indicate relatively high difficulty to green that represents relatively low difficulty. In addition to that, the code segments can be also colored based on a specific AOI metric. Figure 5.5 displays a mockup of this segment-per-segment data visualization approach. In the selection area, the participants that should be considered for the metric computation can be specified.

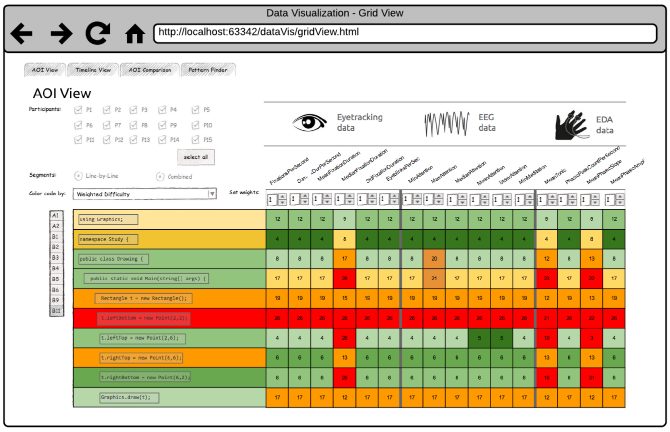


Figure 5.5: Mockup of *Grid View*.

AOI Comparison View. This visualization approach illustrated in Figure 5.6 is closely related to the previously explained view but focuses on the metric data comparison of various code segments from different tasks. The idea behind this approach is to make side-by-side comparisons of code segments across multiple tasks.

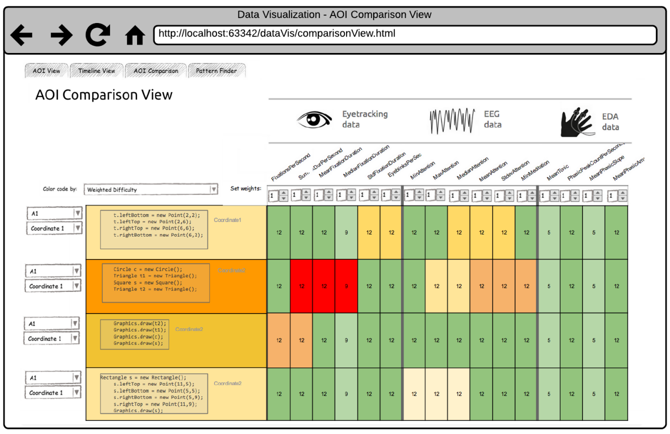


Figure 5.6: Mockup of *AOI Comp. View*.

Timeline View. In the *Timeline View*, data for a specific task and a specific participant (can be selected in the selection area) is visualized. It is intended to display the fixations over the time as *Fixation Bars*, represented as black rectangles (see Figure 5.7). Other measurements, as for example MindBand- or EDA data are visualized as line charts. For reasons of clarity, a toggle function should be implemented that allows to hide specific measurements. In addition to that, the code segments can be colored based on a specified metric as it was shown for the *Grid View*.

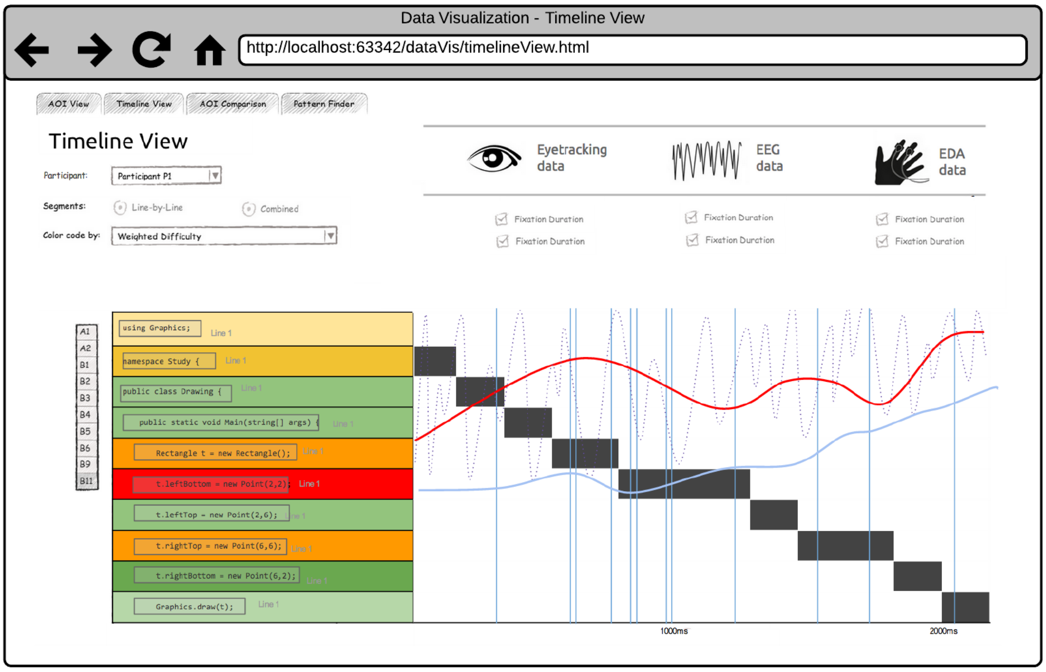


Figure 5.7: Mockup of *Timeline View*.

Aggregated Timeline View. If we talk about time related patterns, a view that allows to compare aggregated time related data is desirable. To discover fixation paths of multiple participants, an approach as presented in Figure 5.8 could be suitable. In this view, the color intensity of each single block represents the number of participants that focused the specific AOI at a specific moment in time. This approach is used to identify code segments with high fixation density at a given moment in time. High fixation density could then be used to locate difficulties a majority of participants experienced for a given time interval.

Figure 5.8: Mockup of *Aggregated Timeline View.*

AOI Scatterplot View. Although the approach shown in Figure 5.9 is not implemented as part of the application, the idea of this AOI comparison approach is explained in short: This view could be used to compare measurement data of comparable AOIs. To allow the comparison over multiple measurements, a scatterplot approach can be choosen. This would allow AOI-specific comparisons between multiple participants in one single view. In Figure 5.9, scatter plot framed in purple on the right hand side of the view shows measurement data of the purple framed segments of the comparable tasks.

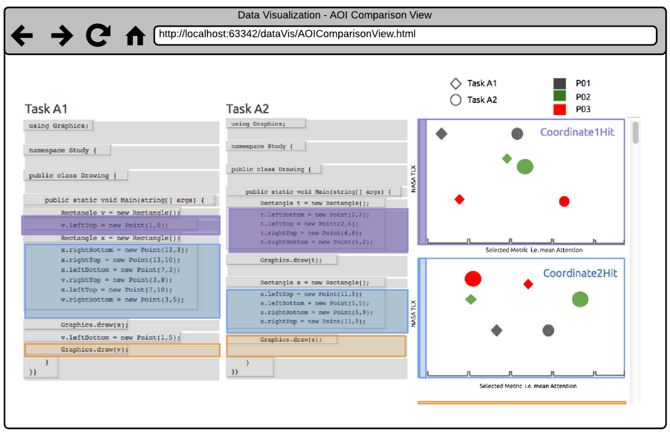


Figure 5.9: Mockup of *AOI Scatterplot View.*

5.3 Application Architecture

This section describes the architecture of the visualization prototype that is implemented as a JavaScript web application. The data visualization approaches are mainly implemented using the JavaScript library D3[1]. Rest based web services are created on a NodeJS[2] server to manage the database access. The NodeJS module named as *node-mysql* allows easy querying for the various web service requests. A list of all created web services can be found in Appendix B. The data used for the visualizations can be categorized into measurement data and task data. The category of task data includes the code of the experimental study tasks as well as the definitions of the areas of interest. These AOI definitions are stored as JSON objects and include *e.g.*, name of area of interest, number of lines or the start line of an AOI. Additionally, for each AOI, it is defined whether it should appear in the line-by-line visualization mode or the combined visualization mode. Currently, the JSON document that contains all AOI information is stored locally. The JSON document containing the AOI definitions as well as the code comprehension tasks as such are stored on a MongoDB database. For the front-end implementation, the module pattern is used. Depending on the view that is selected in the web application, the required modules are loaded. *e.g.*, the `CodeControlModule` that is responsible for displaying the source code of the experimental task on the left hand side of the view is used in all the implemented visualization approaches and is therefore loaded in each of the visualization approaches.

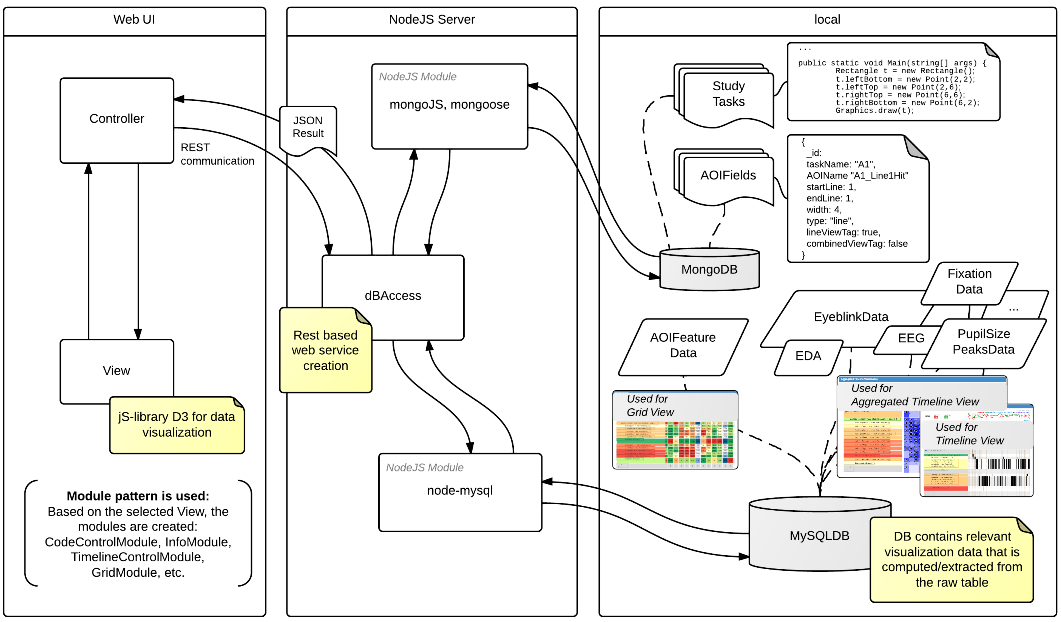


Figure 5.10: Architecture of visualization prototype.

[1]http://d3js.org/
[2]http://nodejs.org/

5.4 Implementation Details

This section gives an overview of the main modules that are defined in the front-end implementation. Each of the three modules that are listed below also represent an implemented view.

Timeline Control Module. The `TimelineControlModule` is loaded if a visualization is requested from the *Timeline View*. Based on the selections made in the multiple dropdown menus of the selection area in the user interface, web service requests are processed in the `DataRetrieveModule`. The retrieved measurement data is then transformed into a suitable format for visualization. This is performed by the `DataTransformationModule`. Like in all the visualization views, the `CodeControlModule` is loaded based on the selected comprehension task. It displays the code on the screen, segmented by the areas of interest. If selected, the scan path is drawn additionally based on the fixation data. Visualizing the complete set of measurement data for the timeline (*e.g.*, fixation data, MindBand data, EDA peaks and pupil diameter peaks) is performed in the module named as `LineGraphModule`. Moreover, the `InfoModule` responsible for displaying general task information in a container above the visualization is loaded.

Grid Control Module. This module is responsible for displaying the table-like *Grid View*. For that, computed AOI specific measurement data are retrieved from the database for the selected participants (`DataRetrieveModule`). If multiple participants are selected, the mean/median is computed for each AOI and each AOI metric. Like in the *Timeline View*, the `CodeControlModule` that is responsible for displaying the code comprehension task is loaded. The same applies for the `InfoModule` that presents statistical information about the task.

Pattern Map Module. The `PatternMapModule` represents the Aggregated Timeline View (heat map like approach). Within this module the `CodeControlModule` is loaded as well as the `InfoModule`. The `PatternMapModule` includes also the computation of the number of participants for which hits were recognized for a specific AOI in a given time interval. Similar computations, that are also part of this module are performed for the number of EDA- and pupil diameter peaks per time interval and AOI respectively.

5.5 Data Storage

5.5.1 Tasks and AOI Definitions

The tasks of the experimental study are stored as text files. To display the content of the AOIs in alignment to the measurement data, a method to extract the relevant lines from the task file had to be found. For this, a JSON document containing the AOI definitions was created.

In Listing 5.1, an extract of that AOI definition document is given. It shows the configuration data of two sample AOIs. Using this information, the lines that correspond to a given AOI are extracted from the corresponding study task document.

```
{
    "tasks": [
            {
                "name": "A1",
                "size": 23,
                "question": "Will the two drawn rectangles overlap?",
                "answers": "yes / no",
                "segments": [
                    {
                        "aoi_name": "A1_AOILine1Hit",
                        "aoi_nLines": 1,
                        "aoi_sLine": 1,
                        "aoi_nLinesBefore": 0,
                        "aoi_nLinesAfter": 1,
                        "aoi_width": 5,
                        "aoi_type": "line",
                        "lineByLineTag": true,
                        "combinedTag": true
                    },
                    {
                        "aoi_name": "A1_AOICoordinate1Hit",
                        "aoi_nLines": 4,
                        "aoi_sLine": 9,
                        "aoi_nLinesBefore": 0,
                        "aoi_nLinesAfter": 0,
                        "aoi_width": 10,
                        "aoi_type": "coordinate",
                        "lineByLineTag": false,
                        "combinedTag": true
                    },
                    ...
```

Listing 5.1: Extract of JSON document representing configuration details of AOIs.

5.5.2 Measurement Data

The relevant measurement data for the visualizations are generated using Matlab computations and database extraction processes. Some of them are used for the *Grid View* (*i.e.*, feature data), whereas other tables are used for all visualization approaches (*i.e.*, task information data). A detailed overview of the tables used by the prototype was given in Figure 4.4 in the previous chapter.

Chapter 6

Tool Features and Use Cases

In this chapter, the main features of the implemented visualization application are presented in a variety of use case scenarios. First of all, an overview of the user interface components and the available visualization screens is given. Then, the implemented approaches, namely *Grid View*, *Timeline View* and *Aggregated Timeline View* are presented in detail.

6.1 Overview of User Interface

The visualization screens consist of three main components that are described below:

- **Selection Module:** In the selection module on the top of the user interface, an experimental task as well as the participant(s) that should be considered can be selected. As soon as the *Visualize*-Button is pressed, data requests are performed based on the selections. Then, the required modules are loaded and the visualization appears on the screen.
- **Information Module:** In this module, information data about the selected task and the selected participant(s) is displayed. This includes *e.g.*, the task duration, the order of tasks, the NASA TLX rating or the information whether the task is solved correctly or not.
- **Visualization Module:** The visualization module represents the data visualization as such. Detailed explanations about each of the views is given in the following sections.

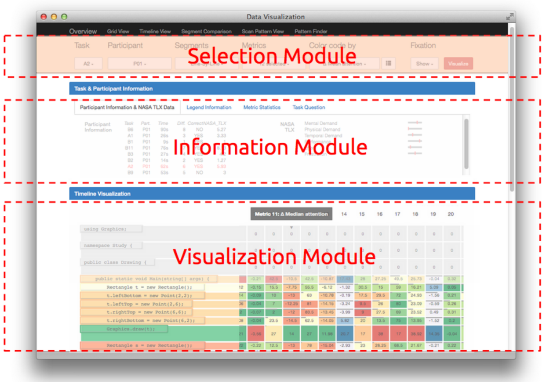


Figure 6.1: Overview of user interface components.

Table 6.1 gives an overview of the implemented visualization views. The basic intention behind each single approach is already explained in brief with corresponding mockups. In the following sections, each view is presented in various use case scenarios. The scenarios that are presented are listed in the last column in the Table 6.1.

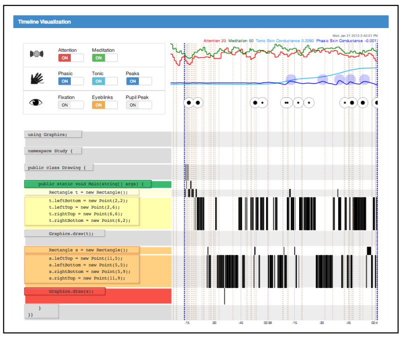
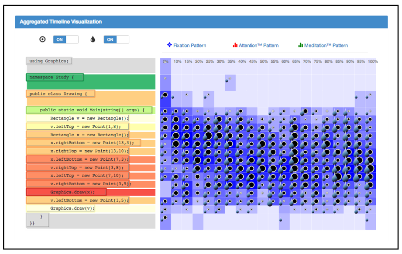
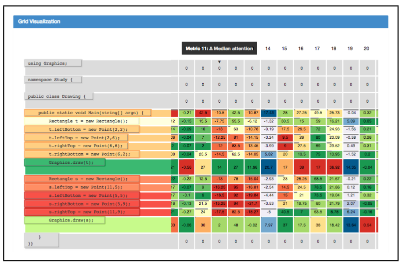
Visualization Module	Description	Scenarios
	**Timeline View** The measurements are displayed on a timeline for a selected task and a specific participant. Using the highlighting feature, pupil diameter data or EDA peaks can be related to specific AOIs.	**Scenario 1:** Identify eye movement patterns **Scenario 2:** Analyze EEG, EDA and pupil diameter data **Scenario 3:** Extract AOI related measurements
	**Aggregated Timeline View** In this approach, multiple participants for a specific task can be selected. It is a heat map like approach providing fixation density information and EDA- and pupillometry data.	**Scenario 4:** Analyze aggregated fixation data **Scenario 5:** Analyze aggregated Attention/Meditation data
	**Grid View** Measurement data that was recorded while the subject focuses on a specific AOI is taken into account for the AOI metric calculation. Multiple participants can be selected to retrieve aggregated data.	**Scenario 6:** Compare AOI metrics **Scenario 7:** Compare fixation paths

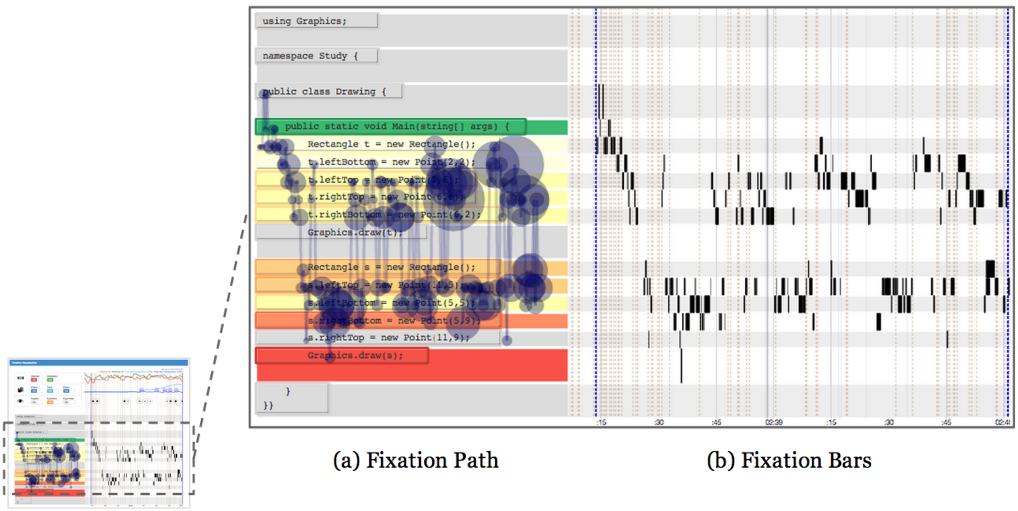
Table 6.1: Implemented visualization approaches with corresponding use case scenarios.

6.2 Timeline View

6.2.1 Scenario 1: Identify Eye Movement Patterns

The first scenario focuses on eye movement patterns. In the experimental study, the gathered gaze events are saccades and fixations. Since only fixations can be related to certain AOIs (in contrast to saccades), only the fixation data is relevant for the visualization. Figure 6.2 shows an extract of a sample timeline visualization. On the right hand side, the line-by-line fixation data is displayed in a similar way as it is shown by Uwano *et al.* in the form of *Fixation Bars*. In addition to that, the fixation data is directly included into the code area on the left as a *Fixation Path*. In the latter approach, the size of the node is representing the fixation duration. To identify high fixation density the fixation path approach is better suited. An analysis of the visualization example in Figure 6.2 gives the following insights:

- The participant scans the code four times.
- In the first few seconds of the task, the duration of each fixation is relatively short.
- The scan time of the first scan amounts to 15 seconds.
- In some of the AOIs no fixation is recognized.
- An accumulation of eyeblinks (brown dashed lines) can be identified at the beginning of the task and in the last few seconds of the task.
- As the background color of the AOIs represent in this case the metric mean attention. So it can be said that the mean attention of the AOIs in the lower part of the code is higher than in the first few lines.



(a) Fixation Path (b) Fixation Bars

Figure 6.2: Fixation data visualization approaches: (a) *Fixation Path* in code area (size of nodes indicates fixation duration), (b) *Fixation Bars* in measurement area (width of bars indicates fixation duration); task A2, participant P01.

6.2.2 Scenario 2: Analyze EEG, EDA and Pupil Diameter Data

In this scenario, the focus is set on the analysis of EEG, EDA and pupil diameter data for a single particpant. An extract of a sample *Timeline View* with focus on these three measures is depicted in Figure 6.3. For clarity reasons, the measures are displayed in a separate lane. The first lane shows the Neurosky's eSense™ values (a) Attention and Meditation as a line chart. The second lane consists of the (b) phasic and tonic component of the EDA-signal whereas the peaks in the phasic component are highlighted with blue circles. In the lowest lane, (c) pupil diameter peak data is displayed using pupil icons that are placed on the timeline. The size of the inner circle in black represents the deviation of the pupil diameter peak to the mean pupil size of the dominant eye during the task. Additionally, the (d) eyeblinks are represented by dashed vertical lines.

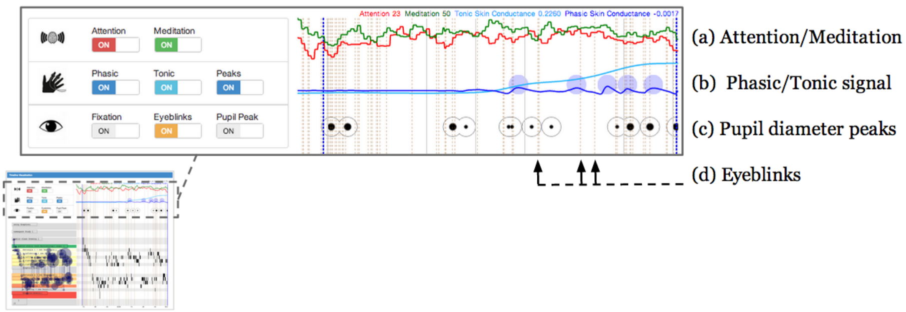


Figure 6.3: Extract of *Timeline View* with visualized measurements (Attention/Meditation, EDA peaks and pupil diameter peaks).

Using the highlighting feature integrated into the timeline, it is possible to hover over the icons of pupil diameter peaks as well as those of the EDA peaks to highlight the AOIs that can be associated with it. This peak hovering feature is illustrated in Figure 6.4. Based on literature, the latency amounts to 1 to 3 seconds for SCRs [Chr81]. This means, that all fixations that happened in this time range are probable reasons for a stimulus. On the right hand side of Figure 6.4 (b), the related AOIs for a EDA-peak are highlighted whereas on the left hand side (a), the code segments that can be related to the selected pupil diameter are colored.

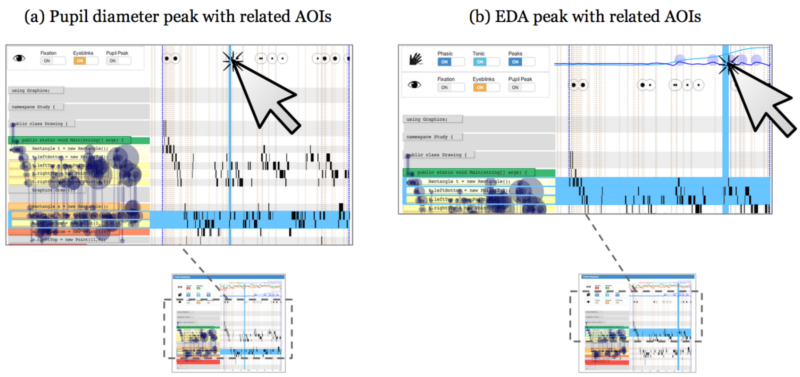


Figure 6.4: Illustration of peak highlighting feature: (a) pupil diameter peak and (b) EDA peak.

6.2.3 Scenario 3: Extract AOI Measurement Data on Timeline

This scenario presents the highlighting feature for AOI specific measurements. As Figure 6.5 illustrates, a click on a specific AOI highlights the fixations in the selected AOI. This allows *e.g.*, to identify how often a participant focuses a specific line in an easy way. In addition to that, the corresponding measurement data in the three lanes on the top of the visualization are highlighted as well. This allows to identify data that has been recorded while the participant focused on a specific AOI more precisely. The following findings can be extracted from the visualization in Figure 6.5:

- The participant focuses the selected AOI basically four times for a longer period of time.
- One EDA-peak is noticed while the participant has been focused on the selected AOI.
- A few pupil diameter peaks that can be related to the selected AOI can be detected in the last few seconds of the task.

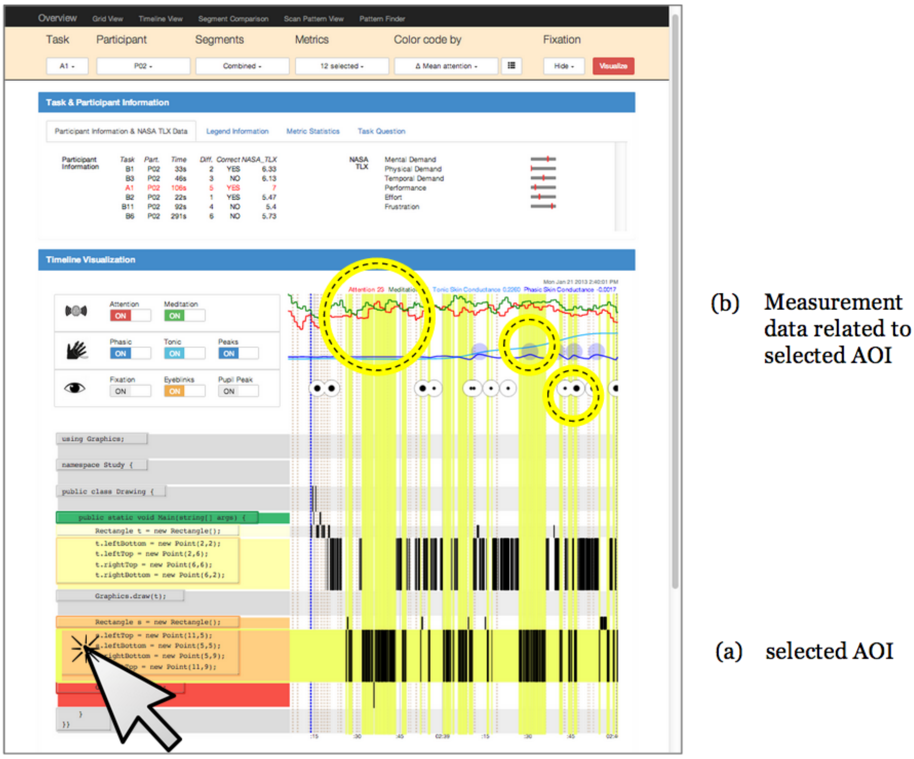


Figure 6.5: Illustration of AOI highlighting feature in the *Timeline View*.

6.3 Aggregated Timeline View

6.3.1 Scenario 4: Analyze Aggregated Fixation Data

Because the previously presented timeline approach do not allow to visualize task data for multiple participants at once due to clarity reasons, an *Aggregated Timeline View* in form of a heat map like approach is implemented. An extract of a sample output is given in Figure 6.6. This approach can be used to identify high fixation densities at a given time point during the task. The more participants fixated a specific AOI during the time unit a block represents, the more intense the blue appears on the screen for the corresponding block. In addition to the fixation density, EDA peaks as well as pupil diameter peaks that were found for the selected participants were integrated into this view. The pupil icon represents pupil peaks, the drop of water represents EDA peaks. The larger the icon the more peaks were recognized while the participants fixated a specific AOI at a given time point. The approach supports to display data in a scaled mode (1 block = 5% of task time) as well as a non-scaled mode (1 block = 1 second of task time).

In the following, the aggregated timeline visualization for task B11 (scaled version), depicted in Figure 6.6, is described: It can be said that 9 participants fixated the initialization line of the object array in the time interval from 5 to 10 percent of the individual task duration. Because this is relatively a high number of participants, the corresponding block appears in a slightly darker blue than blocks for other code segments in the same time interval. Additionally, a high number of pupil diameter peaks can be noticed in the first 5% of task time, indicated by the relatively large *pupil icons*. The *drop icon* represents potential stimuli that can be related to EDA peaks. The larger the *drop icon* appears in a specific block, the higher the number participants who potentially had encountered an external stimulus (indicated by EDA peaks) in that specific time slot and that specific AOI.

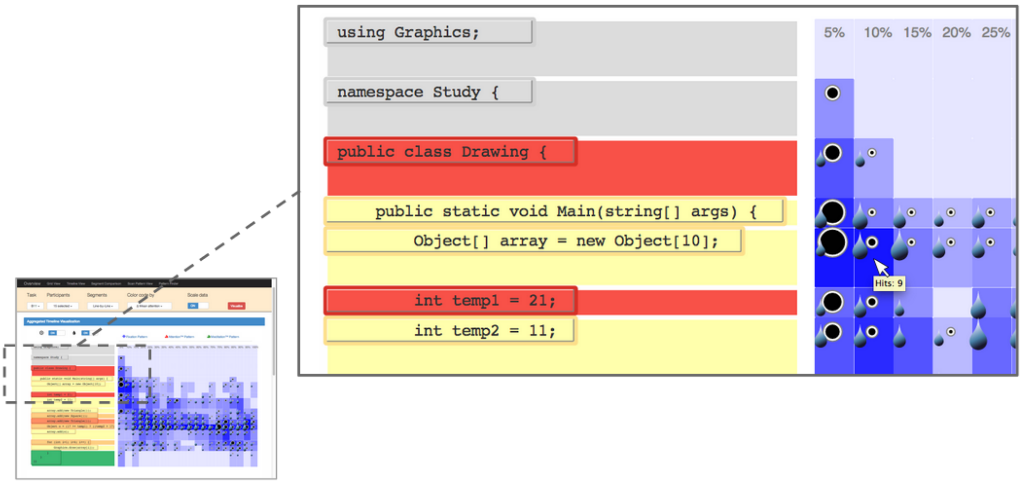


Figure 6.6: Extract of *Aggregated Timeline View* (task B1, all participants are selected).

Using the toggle switch in the selection bar, it can be selected whether the scaled version or the non-scaled version of the timeline should be shown. An example of the non-scaled version is depcited in Figure 6.7 on the left hand side (only first 20s are shown) whereas the scaled version is shown on the right. In the version on the left, one block represents 1 second whereas the time blocks in the scaled version on the right represents 5% of the total individual task duration.

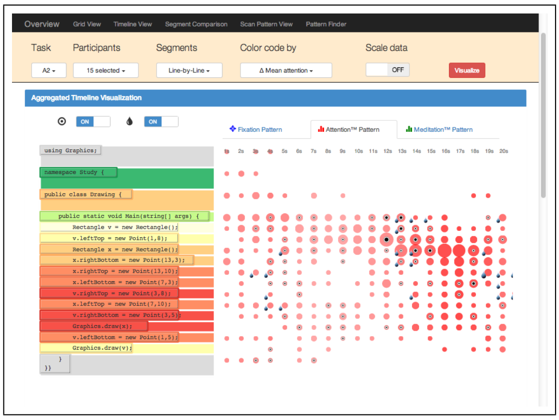
Based on the scaled and non-scaled visualizations given in Figure 6.7, the following statements can be made:

- After about 10% of each participants task duration, most of the participants focused on the while loop in the lower part of the code.
- In the last 5% of each participants' task duration, a slight increase of fixation hits in the first few lines of the code can be recognized. This could imply that participants perform an additional scan of the code short before the answer to the task is given.
- High fixation density in the first four AOIs is mainly recognized in the first 13 seconds of the particpants' task duration.
- Most of the pupil diameter peaks as well as the EDA peaks occurred after 25-95% of the total individual task time.

Figure 6.7: *Aggregated Timeline Views* with focus on fixation data: non-scaled fixation view and scaled fixation view (task B6, all participants are selected).

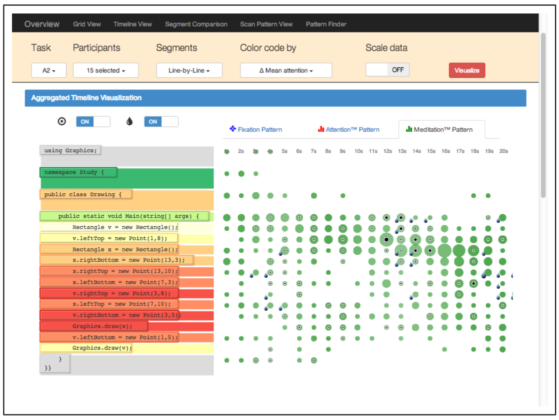
6.3.2 Scenario 5: Analyze Aggregated Attention/Meditation Data

In this scenario, the fixation/attention heat map that can be accessed by the tabs *Attention Pattern* and *Meditation Pattern* in the *Aggregated Timeline View* is used for the analysis. In Figure 6.8, the corresponding scaled and non-scaled versions for task A2 are shown. Basically, it can be said that the view visualizes the aggregated mean attention/meditation values for the selected participants over the task duration. The size of the circles represent the number of participants that fixated the corresponding AOI in the specific time interval, while the color intensity represents the mean attention value. Like in the previous scenario, EDA peaks and pupil diameter peaks are integrated into the visualizations with corresponding icons.

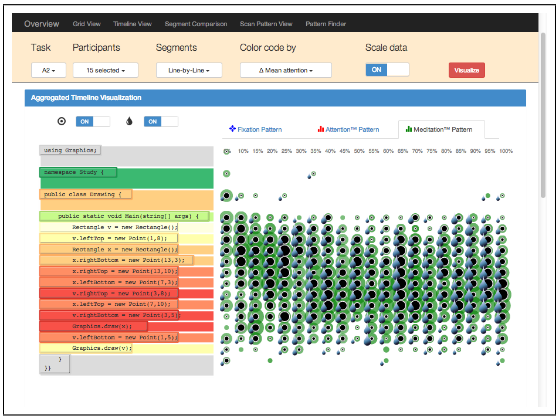


(a)

(b)



(c)



(d)

Figure 6.8: *Aggregated Timeline View* with focus on attention and meditation data: (a) non-scaled attention view, (b) scaled attention view, (c) non-scaled mediation view, (d) scaled meditation view (task A2, all participants selectd).

6.4 Grid View

6.4.1 Scenario 6: Compare AOI Measurements

The intention behind the *Grid View* has already been explained in short with mockups. In essence, it shows the various AOI measurement data in a table-like structure and colors the cells based on their values. Figure 6.9 shows a sample visualization for task A1. For the computation of a specific cell, all the measurement data that was recorded while the selected participants focused on the corresponding AOI is taken into account. Based on the computed values, the cells are colored on a red to green color scale (red indicates relatively high difficulty, green represents relatively low difficulty). Each column represents a metric whose name appears on hover over a column. The subjects that should be considered for the mean metric calculation can be selected in the multiple select dropdown menu that can be found in the selection bar.

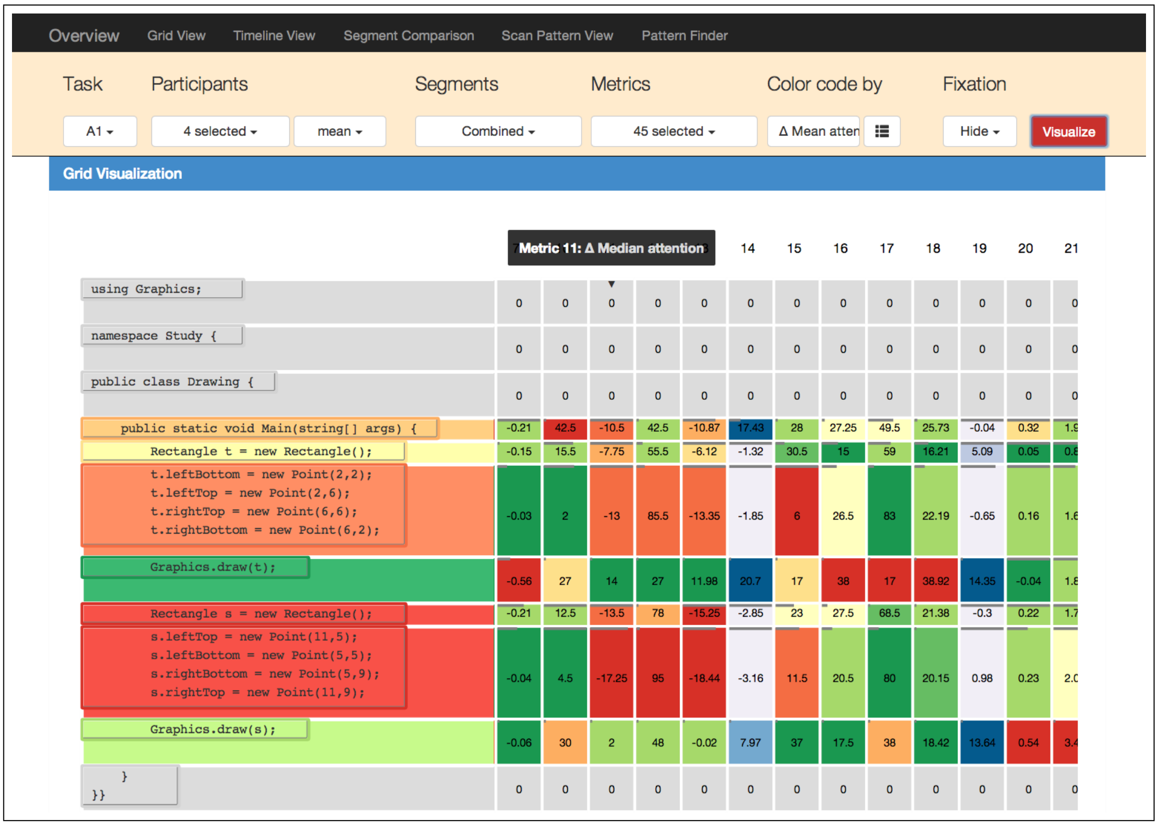


Figure 6.9: Illustration of *Grid View* (task A1, 4 participants selected).

The hovered column (3th column) of the visualization table shown in Figure 6.9 represents the median AOI attention values, respectively the difference of the median attention values (related to the specific AOI) to the baseline data. Based on the illustrated example, it can be said that the median attention values of the field assignments in the second part of the code (colored in red) is relatively higher than the corresponding values of the field assignments in the first part (colored in orange) of task A1.

6.4.2 Scenario 7: Compare Fixation Paths

Since the *Grid View* allows to select multiple participants, the drawing of fixation paths is also supported for multiple participants. This allows to compare fixation data for participants. Although each participants' fixation path is colored differently, a comparison between more than 4 or 5 participants is not recommended due to clarity loss. Figure 6.10 shows an extraction of a sample output. The visualization indicates that for participant P12 a higher number of fixations can be recognized in the loop statement. In contrast to that, participant P15 (red path) shows more fixations in the first part of the code. Additional statements about the fixation durations can be made based on the size of the nodes and the background color of the code area. In this case, each AOI is colored based on its mean fixation duration (red = relatively long fixations, green = relatively short fixations). Further time-related statements are only partially possible because of the constant horizontal spacing between each node.

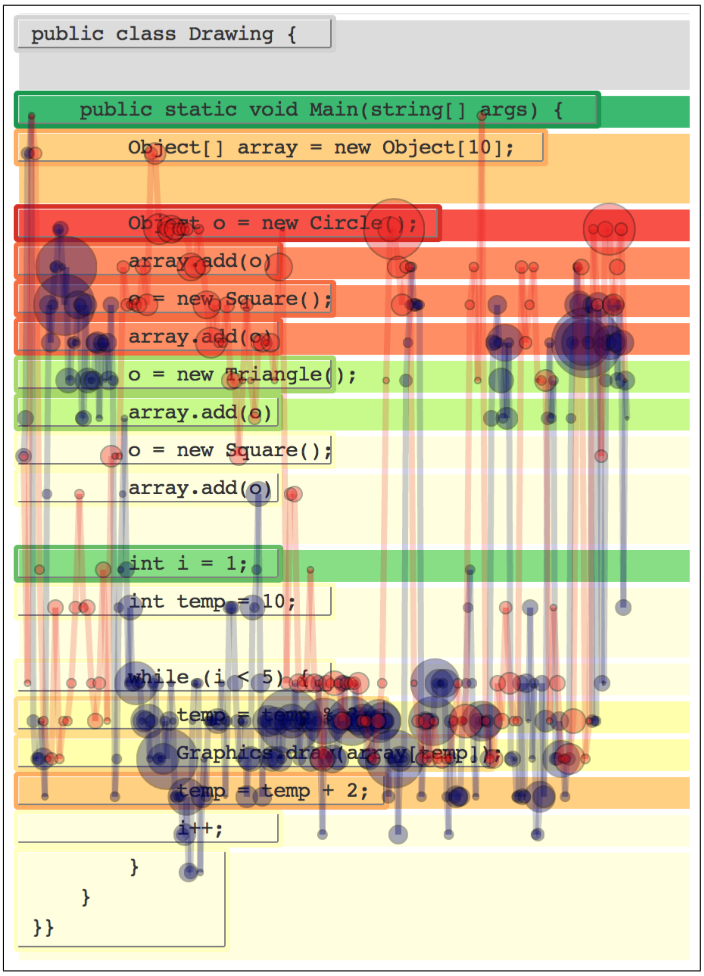


Figure 6.10: Fixation paths of participant P12 (blue path) and P15 (red path), task B6.

Chapter 7

Data Analysis and Pattern Finding

In this chapter the the analysis of the given dataset using the implemented visualization approaches is presented. First of all, the pattern finding approaches are presented in brief. After that, the findings related to various kind of patterns (*e.g.*, code reading patterns, time-related patterns, participant specific findings) are described in several sections. Finally, the results of the statistical tests are summarized.

7.1 Methodology

In order to visually analyze the data in a formal way, screenshots are captured of all 116 *Timeline Views*. All these visualization outputs are then arranged in a table-like structure that allows to inspect and compare the data in an easy way (by task or by participant). This table-like arrangement of the *Timeline Views* is used in all the conducted pattern finding approaches. Additionally, if appropriate, the *Grid View* as well as the *Aggregated Timeline View* is also taken into account. Below, the set of pattern finding approaches is listed:

Fixation Data Analysis using Timeline View. Using the captured screenshots, the fixation data are compared with each other. In addition to that, investigations concerning existing fixation patterns as for example the *scan pattern* or the *retrace declaration pattern* are performed.

Participant-specific Analysis based on Timeline View and Mean Metric Values. To get an overview of the measurement data, mean metric values for each of the tasks are computed. Using bubble charts, a participant-specific analysis is performed regarding aggregated attention/meditation data and time normalized skin- and eye related data. For example, it is examined how the fixation rate or the eyeblink rate differ by participant.

Time-related Analysis using Timeline View and Aggregated Timeline View. The aggregated *Timeline View* is implemented to support the identification of time-related patterns. Based on the findings by visually inspecting the individual *Timeline Views*, the *Aggregated Timeline View* is used to prove whether the findings apply to the majority of participants or not.

Comparable Task Pair Analysis using Timeline View and Grid View. The main goal of this analytical approach is to figure out whether it is possible to explain the difference in difficulty by comparing the *Timeline Views* of task pairs (tasks that had been specifically designed to provide different difficulty level) and evaluating specific AOI measurement data.

Task-property Analysis using Timeline View. In a further approach, all 116 individual tasks of the experimental study are categorized based on various properties. For example, the tasks are grouped into two different groups based on their NASA TLX score. It is then elaborated what kind of differences between the two groups exist (e.g. regarding EDA peaks in the first 20s).

Metric Dependency Analysis using Timeline View. This approach focuses on possible correlations between psycho-physiological measures within task data. In particular, it is analyzed whether any correlations can be identified between measures as for example attention level and EDA peaks or between the fixation rate and the eyeblink rate.

7.2 Fixation Path Patterns

7.2.1 Scan Pattern

The *scan pattern* describes the initial reading of the code, before focusing on specific parts. An analysis of the fixation paths by visual inspection has shown that especially for task A1 and A2 a scan over the entire code is performed multiple times. In contrast to that, in tasks of type B, the scan pattern is rarely visible more than once. A probable reason for that can be found on the tasks itself: In the tasks B1 to B11, the order of the drawn shape objects is asked. There, basic parts in the first few lines of the code do not require to be viewed additionally. In contrast to that, some participants had to read in task A1 and A2 (tasks that are about overlapping rectangles) the entire code multiple times to find and extract the relevant coordinate information. Figure 7.1 shows an extract of a sample *Timeline View* showing two scans.

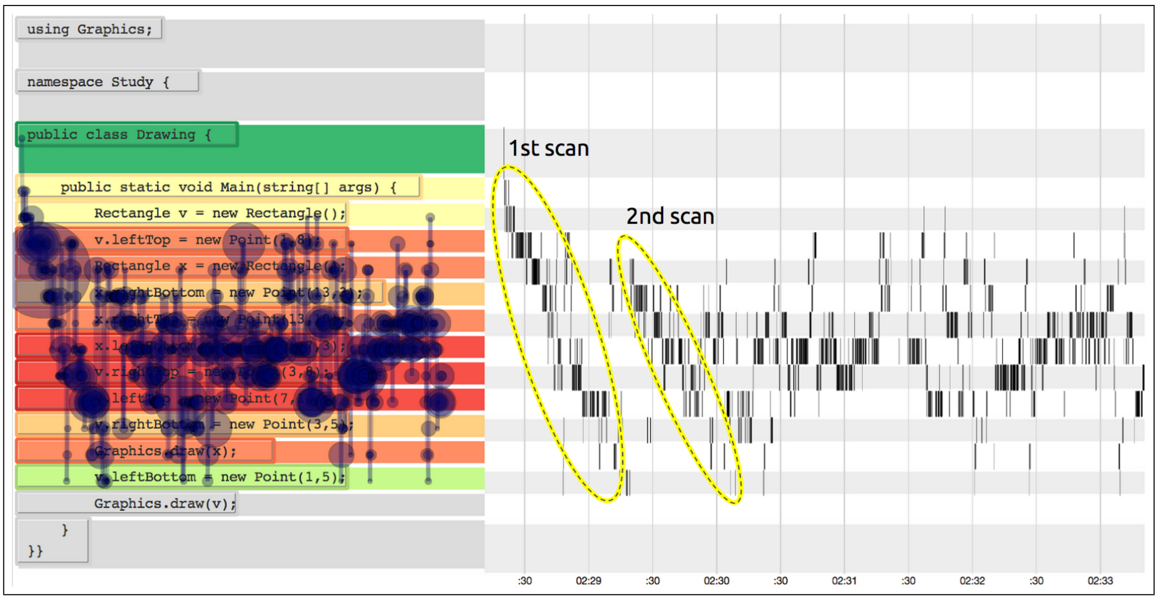


Figure 7.1: Fixation visualization showing two *scan pattern* instances (task A2, participant P07).

Uwano *et al.* defines the *first scan time* as the time until 80 percent of the total lines of code (except blank lines) are read [UNMM06]. Based on the *fixation bars* in the timeline visualization, the first scan time is elaborated for each of the tasks in the dataset under consideration. Results of various code review studies have shown that developers who spend more time for the initial scan tend to use less time for error detection (*e.g.*, [UNMM06], [SFM12]). In this thesis a similar analysis is conducted. In contrast to the code review studies by Uwano *et al.* and Sharif *et al.*, no significant correlation between the first scan time and the task completion time was found; see box on the right.
An additional statistical test that uses the correctness of the tasks as performance indicator was conducted. Because of the ratio of false and correct answers to a specific task, only task B3 and B11 are considered for this analysis. In both cases, no significant difference was obtained; see middle box on the right.
Additionally, a Pearson correlation coefficient was computed to assess the relationship between the first scan time and the subjects' perceived difficulty rating of the corresponding task. As the dot plot in Figure 7.2 already indicates, there was a positive correlation between the two variables. It was found that for tasks that are perceived as more difficult (based on the participants' rating), a significantly longer scan time was observed; see bottom box on the right.

Pearson's Test:
H_0: *Task completion time is the same regardless of the first scan time (Note: only correctly solved tasks were considered).*

There was no correlation between the first scan time and the task completion time. For task B11: [r = -0.453, n = 7, p = 0.221].

Cannot Reject H_0

T-Test (independent samples):
H_0: *Task performance (correct/false answer) is regardless of scan time (Note: performed for task B3 and B11 only).*

There was not a significant difference in the first scan time for correctly solved tasks (M_{B3}=25, SD_{B3}=17.1) and not correctly solved tasks (M_{B3}=26.7, SD_{B3}=13.7); $t_{B3}(12)=0.195$, $p_{B3} = 0.848$

Cannot Reject H_0

Pearson's Test:
H_0: *Scan time is regardless of perceived difficulty ranking.*

There was no correlation between the two variables [r = 0.393, n = 106, p < 0.001].

Reject H_0

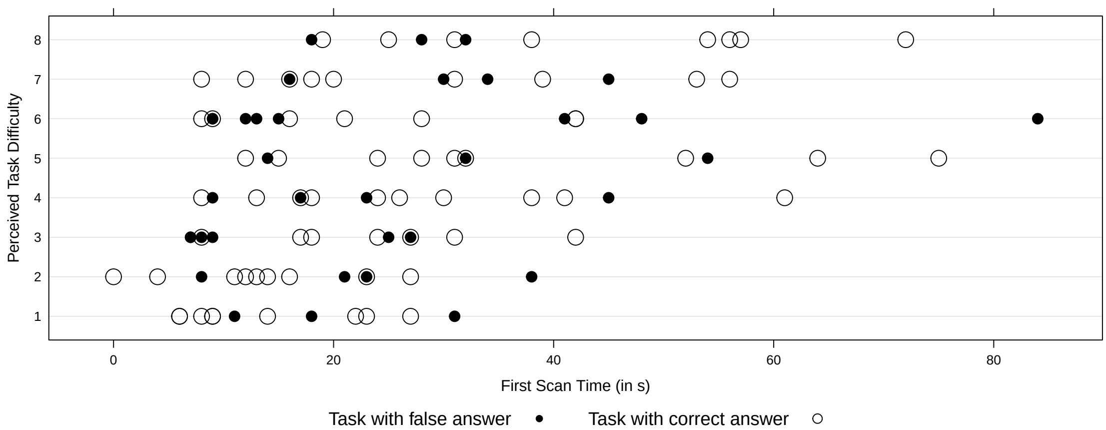


Figure 7.2: Scatter plot that shows the first scan time in relation with the perceived difficulty of the tasks (Note that tasks were omitted because of incomplete eye-tracking data).

7.2.2 Retrace Declaration Pattern

The *retrace declaration pattern* that is presented in the study conducted by Uwano *et al.* is also found in the dataset that is used in this work [UNMM06]. Especially in cases where generic variable names are used as for example in task B2, a significant amount of *retrace declaration patterns* can be identified. An extract of a sample *Timeline View* where the mentioned pattern can be identified multiple times is given in Figure 7.3. In this task, the participant looks back to the variable declaration a few times to be sure what kind of shape object is drawn on the screen.

In task pair B1 - B2, the difficulty level is increased by using generic variable names (naming not related to variable) instead of mnemonic variable names (aims at retaining what a variable stands for). To quantify the effect of using mnemonic variable names, a comparison regarding the number of retrace declaration pattern instances seems to be an appropriate approach. Such an approach is performed in the comparable task pair analysis; see Chapter 7.5;

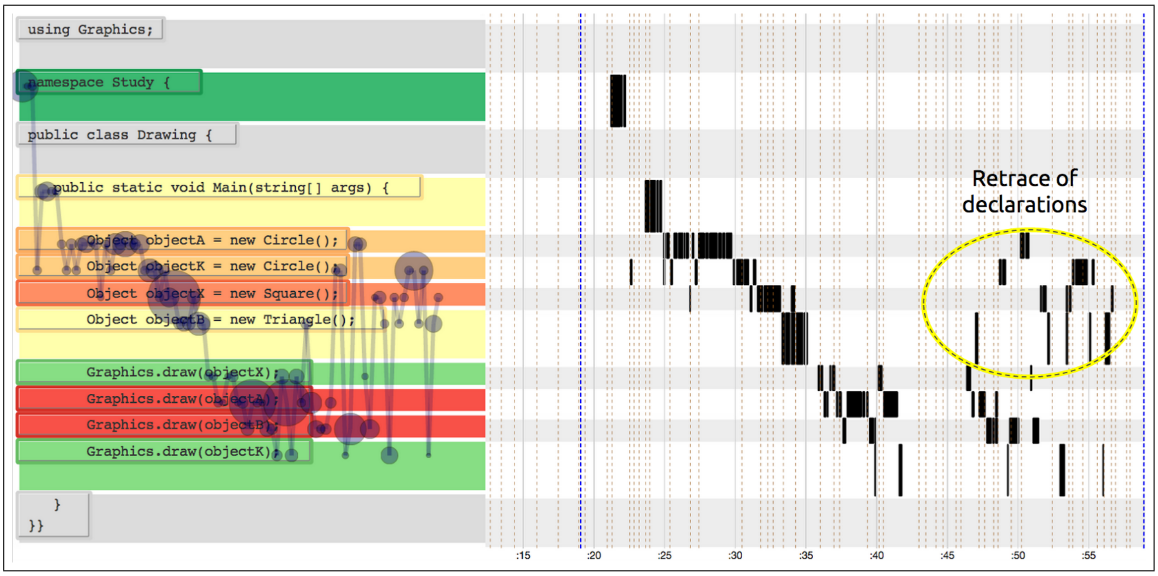


Figure 7.3: Fixation visualization that includes the *retrace declaration pattern* (task B2, participant P13).

7.2.3 Retrace Reference Pattern

When the participant looks back to lines where a specific variable is recently referred, a *retrace reference pattern* is identified [UNMM06]. Due to the way most of the tasks are constructed, no significant amount of retrace reference patterns has been noticed in the given dataset. A few instances can be found in task B6, B9 and B11. Because of the low amount of total instances, no quantitative analysis regarding *retrace reference patterns* can be performed. However, to complete the set of reading patterns defined by Uwano *et al.*, an illustrated example for that pattern is given below [UNMM06].

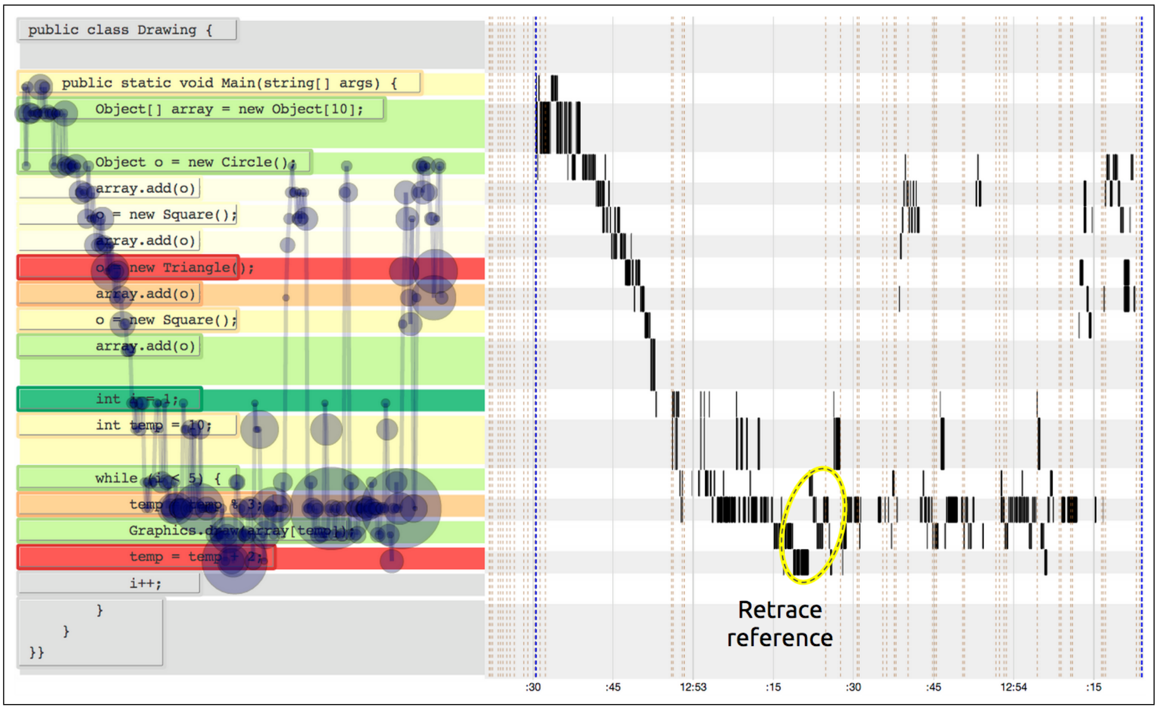


Figure 7.4: Fixation visualization that includes *retrace reference pattern*.

7.2.4 Multiple Scan Pattern

Apart from considering each fixation path in isolation, it is also analyzed how the paths differ from participant to participant and from task to task respectively. In addition to that, it is also elaborated what kind of fixation data differences exist between tasks that were solved correctly compared to tasks where the participants were not able to find the correct answer. Considering fixation data of tasks A1 and A2 grouped by correctness shows interesting insights that are illustrated with a selection of fixation data visualizations in Figure 7.5 and Figure 7.6 (correspond to task A1 or A2 and represent tasks with correct answers and tasks with false answers, respectively). Figure 7.6 shows the fixation paths that correspond to tasks that were solved not correctly, whereas Figure 7.5 shows a selection of fixation visualizations that correspond to tasks that were solved correctly. Scans are highlighted in blue whereas time periods where the participants focused on specific part of the code are highlighted in red. As visual inspection of these visualizations (which represent the majority of each category) indicates, a larger number of scans can be noticed for tasks that were not solved correctly. In most of the tasks that were solved correctly, there are no more than two scans seen before the participant focuses on a specific part of the code. For this analysis, tasks A1 and A2 were considered only because of the small amount of scans identified in the other tasks (due to code structure). An independent-samples t-test was conducted to compare the number of scans for tasks that were solved correctly and tasks that were not solved correctly (considering tasks A1 and A2 only). The result show that there was a significant difference in the number of scans for tasks that were solved correctly (M=1.28, SD=0.843) and for those that were not solved correctly (M=2.75, SD=1.5); see box on the right below.

In this work, this pattern is named *Multiple Scan Pattern* and can be interpreted as a cognitive action of reading the code multiple times without recognizing the crucial parts of the task. In the case of the tasks under consideration (A1 and A2), the participant has to figure out whether two rectangles overlap or not. The crucial parts can be defined as the coordinate definitions of the vertices where an overlap is possible (*e.g.*, coordinates of bottom left corner of rectangle 1 and coordinates of top right corner of rectangle 2).

T-Test (independent samples):
H_0: The number of scans is regardless of the task performance (correct/false). (Note: Tasks A1/A2 are considered only).

There was a significant difference in the number of scans for correctly solved tasks (M=1.28, SD=0.84) and not correctly solved tasks (M=2.75, SD=1.5); $t(27) = 2.908$, $p = 0.007$.

Reject H_0

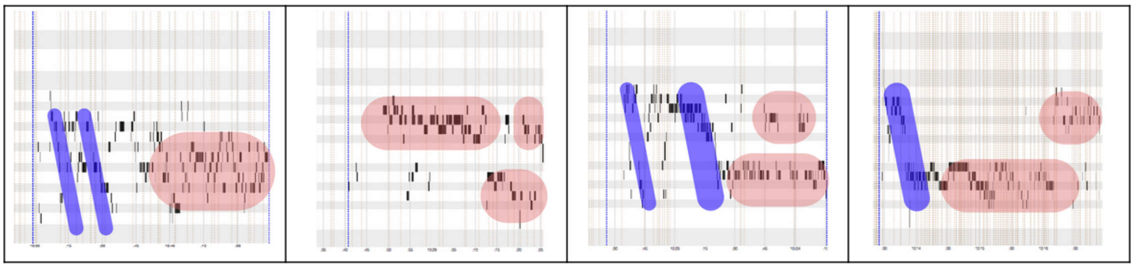


Figure 7.5: Selection of visualized fixations for tasks that were solved correctly (tasks A1 and A2).

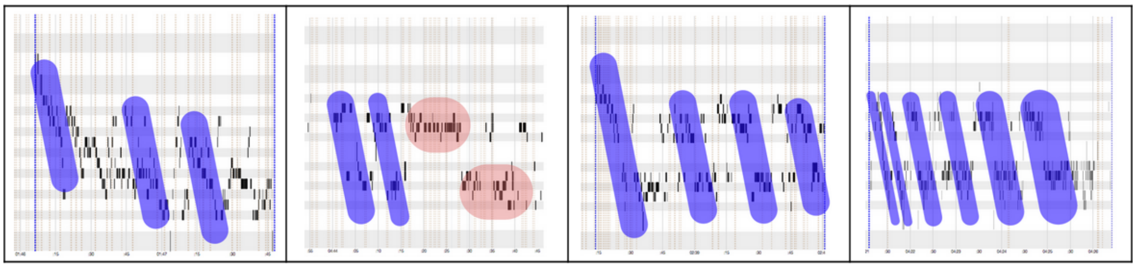


Figure 7.6: Selection of visualized fixations for tasks that were not solved correctly (tasks A1 and A2).

7.3 Participant-specific Insights

7.3.1 Attention and Meditation

In this section, mean task values of NeuroSky's proprietary metrics named *Attention* (mental focus) and *Meditation* (mental calmness) are presented. The bubble charts in Figure 7.7 clearly show that there are participant-specific differences. *e.g.*, the mean attention for participant P01 is for all the tasks below 40, whereas for participant P10 and P11 the lowest value amounts to 48 and 47 respectively. For the meditation metric, differences in a similar range can be noticed.

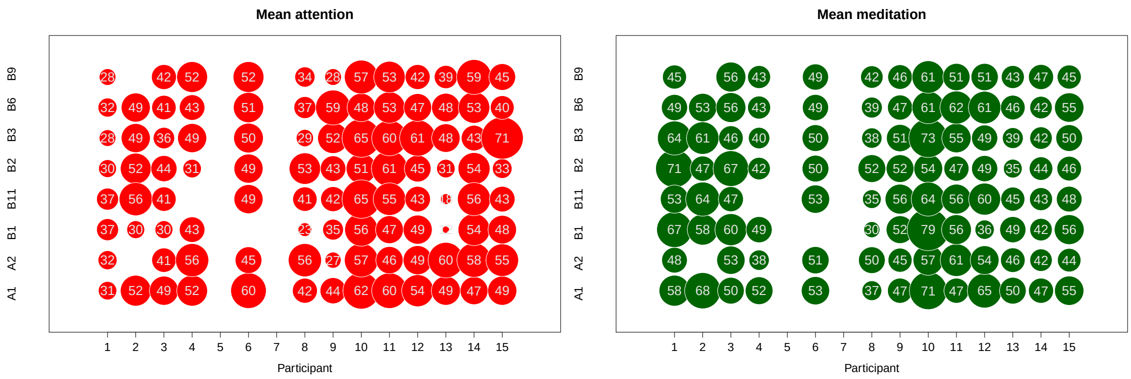


Figure 7.7: Bubble charts representing mean attention and mean meditation level per task.

In order to compare attention and meditation between multiple tasks and multiple participants, normalization should be performed (using recorded data from the second part of the mind relaxation phase/fish tank phase). This normalization is done by subtracting the mean value of the task phase from the mean value of the corresponding mind relaxation phase. Interestingly, participant-specific differences were found regarding the variation of attention data in the said phases. Figure 7.8, however, illustrates that regardless of the task, only three participants (P04, P10 and P11) show consistently higher mean attention during the task phase compared to the mind relaxation phase. For participants P01, P02, P09 and P13, the attention while watching the fish tank video is on average higher than while working on the tasks. This implies that these participants strongly focused on the fish tank video without having wandering thoughts.

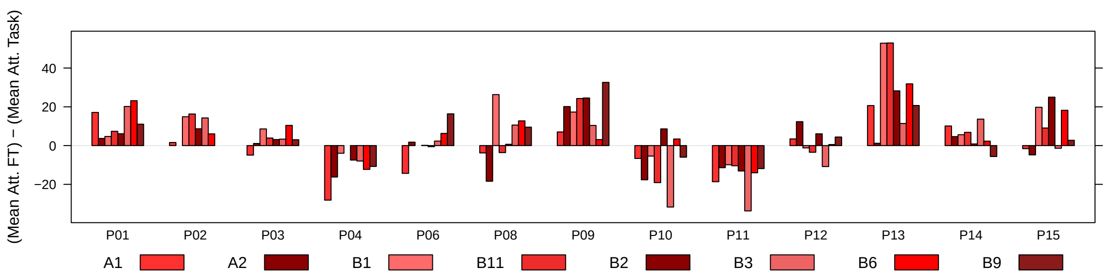


Figure 7.8: Bar chart representing mean attention level per task (normalized), grouped by participant.

7.3.2 Skin- and Eye related Data

Visual inspection of the individual *Timeline Views* indicated that two of the participants tend to provide an extraordinary large number of EDA peak occurrences per time unit (P08 and P11). A sample *Timeline View* for participant P11 is given in Figure 7.9. Apart from the high number of EDA peak occurrences, participant P11 also provides an extraordinary high number of pupil diameter peaks per time unit.

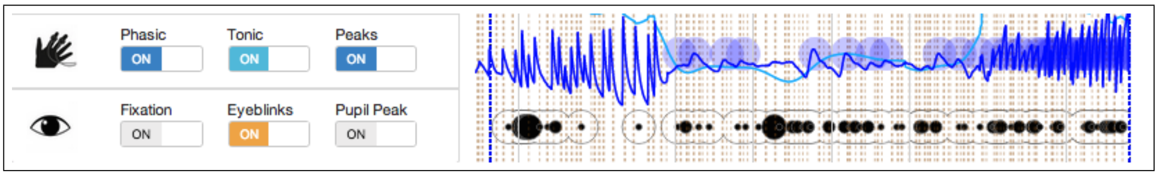


Figure 7.9: Extract of *Timeline View* with high number of eyeblinks, EDA- & pupil size peaks (A2, P11).

The aggregated results of skin- and eye related measures (time normalized per task) are shown in Figure 7.10 as bubble charts. The chart in Figure 7.10a shows that especially for the participants P08 and P11 a high number of EDA peaks is detected. A large number of pupil size outliers per time unit, visible in Figure 7.10b, is provided by participant P11 in three tasks. For the metrics fixation- and eyeblink rate, significant participant-specific differences can be found for participants P14 and P15 (fixation rate) and participants P01 and P06 (eyeblink rate) respectively.

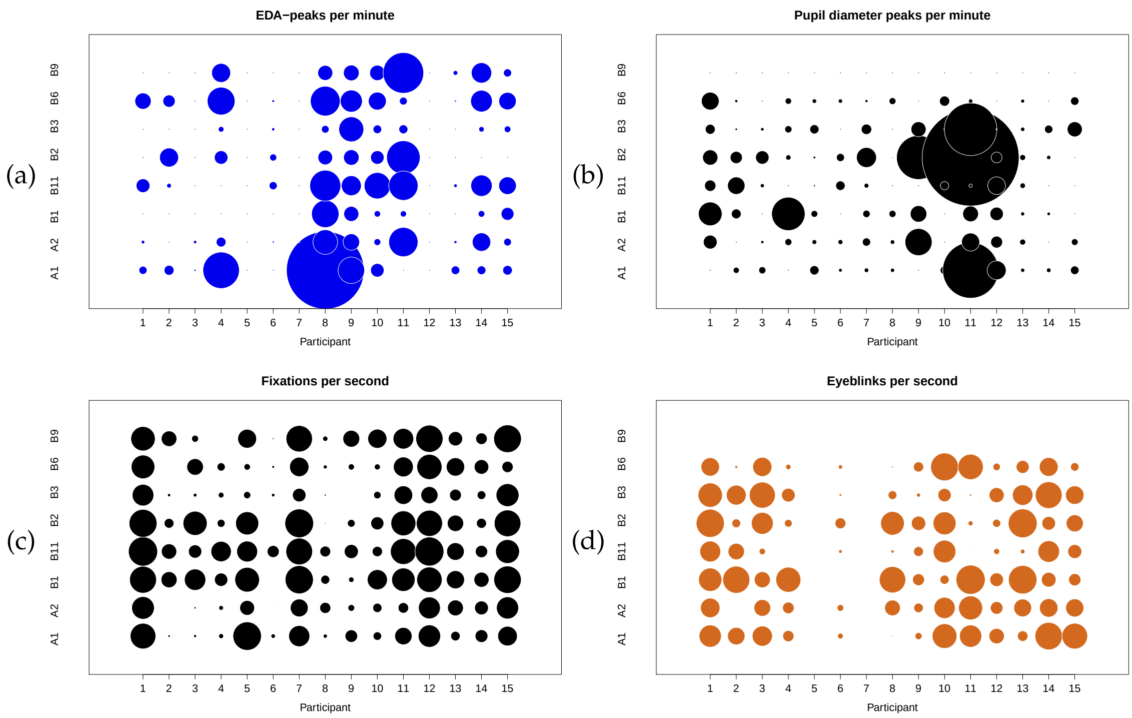


Figure 7.10: Bubble charts representing (a) EDA peaks, (b) pupil size, (c) fixations, (d) eyeblinks per time.

7.4 Time-related Insights

7.4.1 Rising Attention

The *Aggregated Timeline View* allows to analyze how the aggregated attention level as well as the aggregated meditation level vary during a specific task. In a time series analysis of attention data it was found that in 2 of 8 tasks relatively high aggregagted attention values can be observed after about 16-19 seconds. A similar tendency but less clear can be found for two further tasks. In Figure 7.11, the *Aggregated Timeline View* of task A2 that show this phenomena most clearly is illustrated. Note that the more intense the red appears, the higher the mean attention at a given time interval (1s time interval).

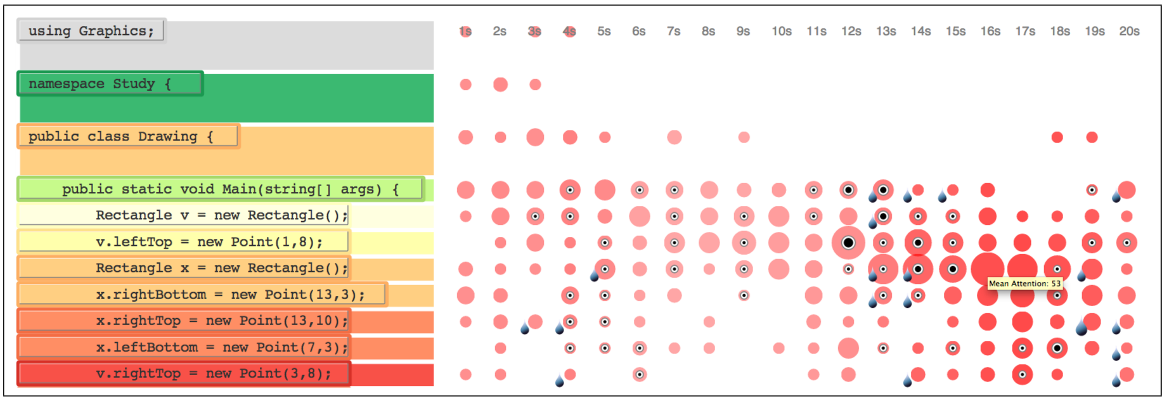


Figure 7.11: *Aggregated Timeline View* for task A2 representing *rising attention*.

The line chart in Figure 7.12 shows the attention level during the first 30 seconds for all the participants who completed task A2. It shows a clear tendency of increasing values but it also shows that the mean attention value is strongly influenced in the start phase by two participants who provided extraordinary low attention values. In summary, although a general tendency of increasing attention is visible in this case, some aspects argue against speaking in terms of patterns in this case.

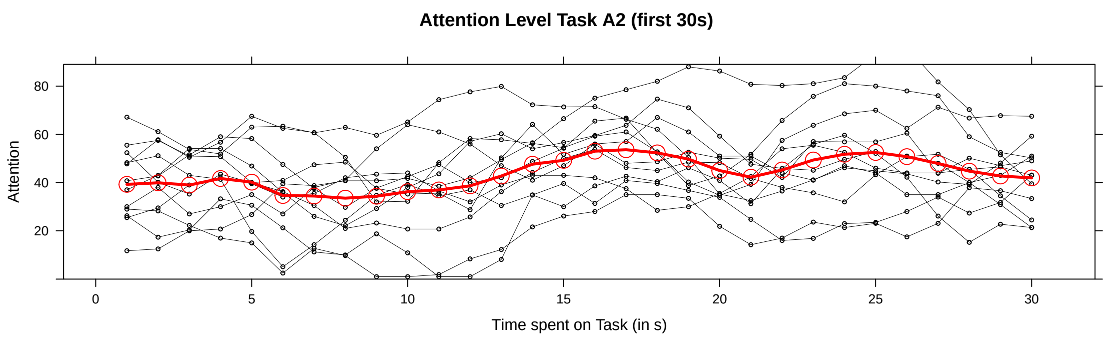


Figure 7.12: Line chart representing participants' varying attention level (first 30s of task A2).

7.4.2 Electrodermal Activity and Pupillometry

An analysis of each single *Timeline View* has shown that in some of the tasks an accumulation of EDA phasic peaks can be found in the last quarter of the task with respect to the time duration. Out of total 60 tasks (where an EDA signal was recorded and the participant took more than 60 seconds), this finding was found in only 10 cases. Six of these are illustrated in Figure 7.13. In additional six cases it was partly shown whereas in rest of the tasks no increase regarding EDA peaks could be observed at the end of the task.

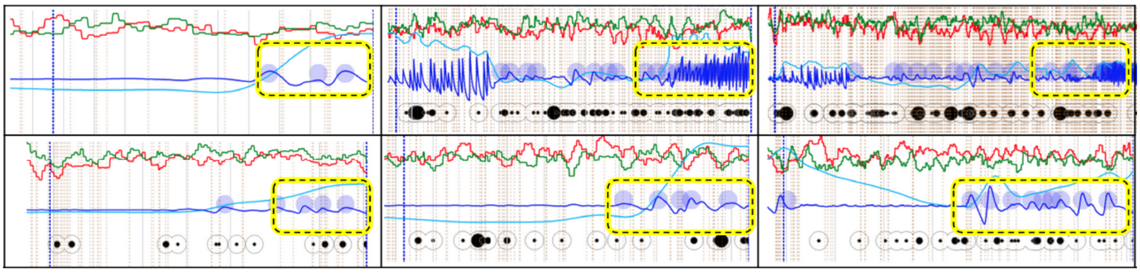


Figure 7.13: Phasic EDA peak locations on timeline show time-related similarities.

Although the *Aggregated Timeline View* seems not to be suitable for a detailed analysis, it can be helpful to prove the described finding. In the scaled mode, it can be analyzed whether for a relatively high number of participants EDA peaks can be noticed in the last quarter of the individual task duration. The larger the drop icons appear on the screen for a given time interval, the more participants provided EDA peaks (which can be related to the corresponding AOI and the specific time interval). An analysis using the *Aggregated Timeline Views* (example in Figure 7.14) for all the tasks suggests that there is no significant time-related difference.

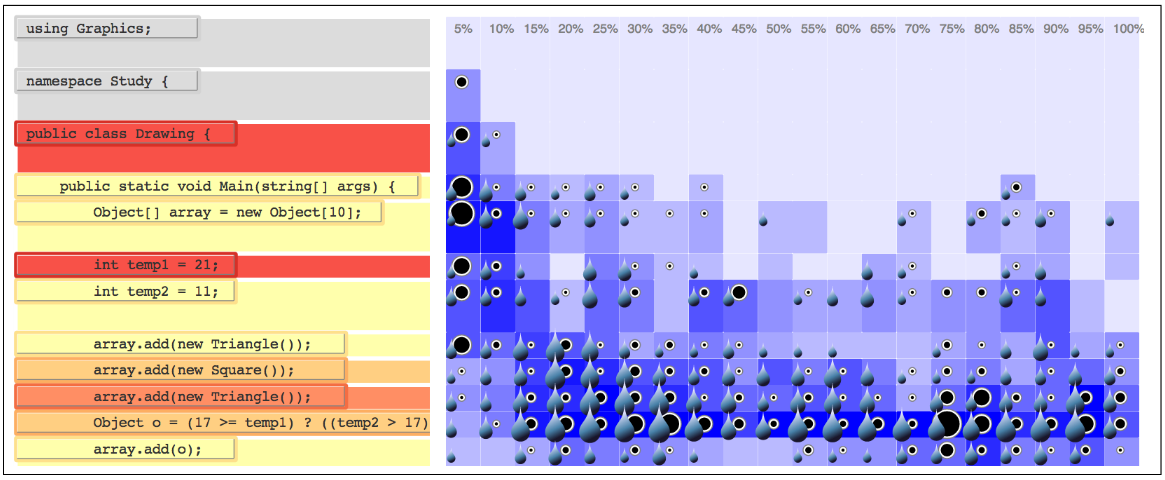


Figure 7.14: Extract of *Aggregated Timeline View* for task B11 (size of drop icons = no. of participants that might have encountered a stimuli that can be related to a phasic EDA peak, size of pupil icons = no. of participants that provided pupil diameter outliers).

7.5 Comparable Task Pair Analysis

7.5.1 Overview of Comparable Task Pairs

Table 7.1 gives an overview of all comparable task pairs and highlights the main differences. In the sections that follow this table, the results of the task pair analysis are presented. For that, similarities or tendencies between multiple participants are elaborated. To clarify the findings visually, sample timeline views are shown for each task pair.

Tasks	Task differences	Description and expected observations
A1 vs. A2	Order of field assignments	In task A1, the fields are assigned in a sequential order whereas the field assignments in task A2 are done randomly. Because of code structure reasons, more fixations are expected for task A2.
B1 vs. B2	Mnemonic vs. generic variable names	For task B2 that involves the generic variable names, a higher number of eye movements between the object definition and the method call is expected because of the vague variable naming.
B1 vs. B3	Use of an array	In task B3, an array is used to store the shape objects. The objects are then drawn by iterating over that array which makes it more complex.
B3 vs. B6	Loop complexity	The loop statement in B6 can be rated as more complex than the one in task B3.
B3 vs. B9	Swap method	In task B9, a swap method is defined that changes the position of the shape objects within the array. Multiple fixations between method call and definition are expected.
B3 vs. B11	Easy loop vs. complex obj. init	In task B11, a very complex line precedes the loop statement. In addition to that, the boundaries of the loop in task B11 can be rated as more complex.
B6 vs. B11	Swap method vs. complex obj. init	The loop statement in task B11 is easier than in B6, but in task B11 there exists a complex line before the loop.
B9 vs. B11	Moderate loop vs. complex obj. init	The loop statement in task B11 is much easier than in B9, but there exists a complex line before the loop.

Table 7.1: Overview of comparable task pairs and expected fixation differences.

7.5.2 A1 vs. A2: Order of Field Assignments

Both tasks, A1 and A2 are about figuring out whether rectangles that are drawn on the screen overlap or not. The rectangles are specified by their coordinates. Figure 7.15 and Figure 7.16 show the *Timeline Views* of the tasks under consideration for a sample participant. In the easier variant of the task (A1), the field assignments are done in a logical order such that the participant can understand quite well which coordinate values correspond to the first rectangle and which belong to the second rectangle. In task A2, the field assignments are mixed up and do not follow any logical order. Because of that, this task is considered to be more difficult and it is expected that the participant takes more time for completing the task. In the visualizations below, the highlighted areas of interest are the coordinate assignments that are crucial to figure out whether there exists an overlap or not.

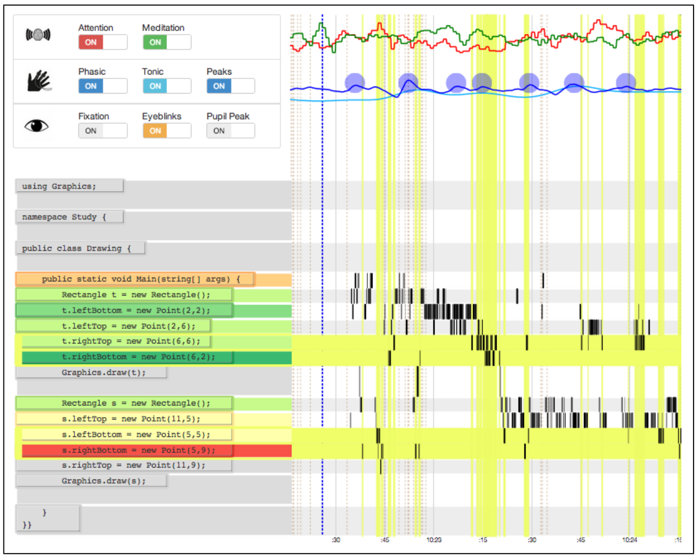


Figure 7.15: Task A1 (participant P09)

Task Duration: ***110s***
Fixations: ***306***
NASA TLX: ***4.6***
Diff. Ranking: ***4***
Correct: ***YES***

Peaks in the phasic part of EDA signal are more or less equally distributed over the complete task duration. No pupil diameter outliers are recognized.

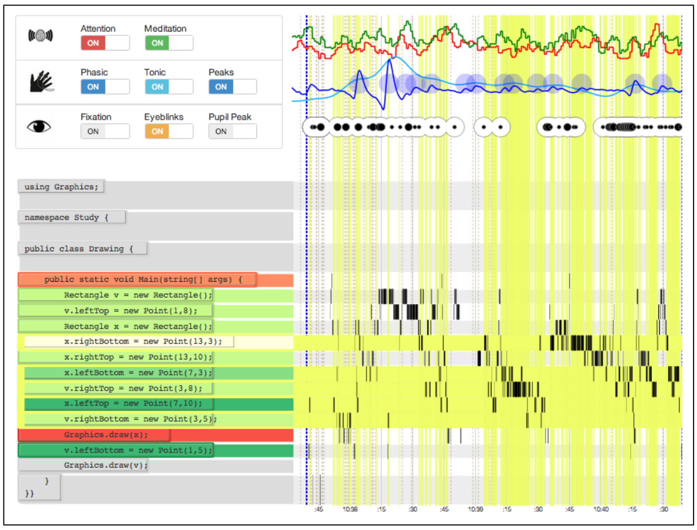


Figure 7.16: Task A2 (participant P09)

Task Duration: ***177s***
Fixations: ***432***
NASA TLX: ***6.1***
Diff. Ranking: ***7***
Correct: ***YES***

High number of pupil diameter outliers can be noticed at the beginning and in the last quarter of the task. Two relatively sharp EDA peaks appear in the first 40 seconds. Task duration is increased by 41% compared to task A1.

Table 7.2 provides mean metric values for the task pair A1-A2. It was found that metrics such as eyeblink rate, fixation rate as well as mean attention and mean meditation do not differ clearly in this task pair. As shown in the illustrated example for participant P09, the number of fixation is increased enormously for task A2. A t-Test was conducted to verify whether the number of required fixations to complete the task is significantly higher if the field assignments follow a logical order. A significant difference was found [t(13) = -2.634, p = 0.021]; see box on the right.

T-Test (paired):
H_0: *The number of fixation is the same regardless whether the field assignments follow a logical order or not.*

There was a significant difference in the number of required fixations for the task where the field assignments follow a logical order (M=267.8, SD=188.1) and the task with field assignments that follow a random order (M=429.6, SD=182.7);
t(13) = -2.634, p = 0.021.

Reject H_0

Task	False Answ.	Time (in s)	NASA TLX	Perc. Diff.	Fixations	Saccades	Fix./sec	Blinks/sec	Pupil P./min	EDA P./min	Attention	Meditation	ΔAttention	ΔMeditation
A1	2	106	8.5	4.4	263	367	2.8	0.45	10.1	3.9	50	54	-1	7
A2	1	166	10.4	6.3	430	587	2.7	0.42	8.6	2.4	49	49	-2	8

Table 7.2: Mean metric values over all the participants who completed task A1 and task A2 respectively.

As seen in the illustrated example in Figure 7.15 and Figure 7.16, participant P09 provided only pupil diameter outliers while working on task A2. But by considering Table 7.2 a higher corresponding mean value (*pupil diameter outliers per time unit*) can be noticed for task A1. As shown in the bubble chart in Figure 7.17, this difference can be explained by participant P11 who provided a huge number of pupil diameter peaks per time unit. However, a participant specific analysis based on the metric *pupil diameter outliers per time unit* has found that no implications are possible for this task pair. The same applies regarding the metric *EDA peaks per time unit*.

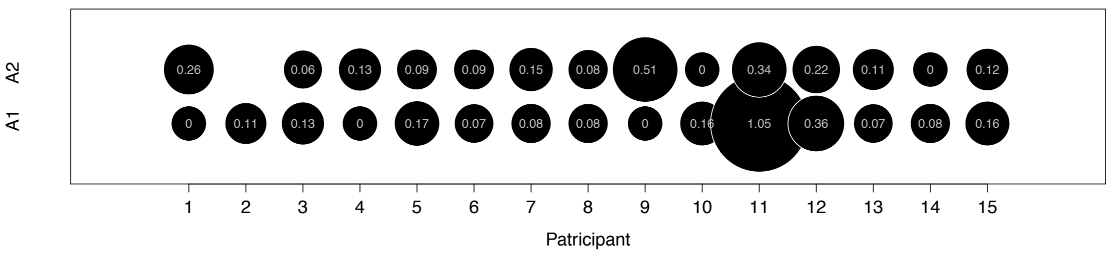


Figure 7.17: Bubble chart representing pupil diameter outliers per second (task pair A1-A2).

7.5.3 B1 vs. B2: Mnemonic vs. Generic Variable Names

In the code comprehension tasks B1 and B2, four geometrical objects are drawn on the screen in a specific order. In task B1, where mnemonic variable names[1] are used, lower perceived difficulty is expected. In Figure 7.18 and Figure 7.19, the *Timeline Views* of the two tasks under consideration are shown for a sample participant. In both visualization, the area of interest that contains the object initialization part is highlighted using the implemented highlighting feature.

Figure 7.18: Task B1 (participant 13)

Task Duration: ***42s***
Fixations: ***128***
NASA TLX: ***5.2***
Diff. Ranking: ***2***
Correct: ***YES***

Fixation duration when participant first focuses the object Initialization segment is about 5 seconds. Participant fixates that AOI a few more times, but all these additional fixations on this segment took less than 1 second each. A pupil diameter peak at the beginning of the task is recognized.

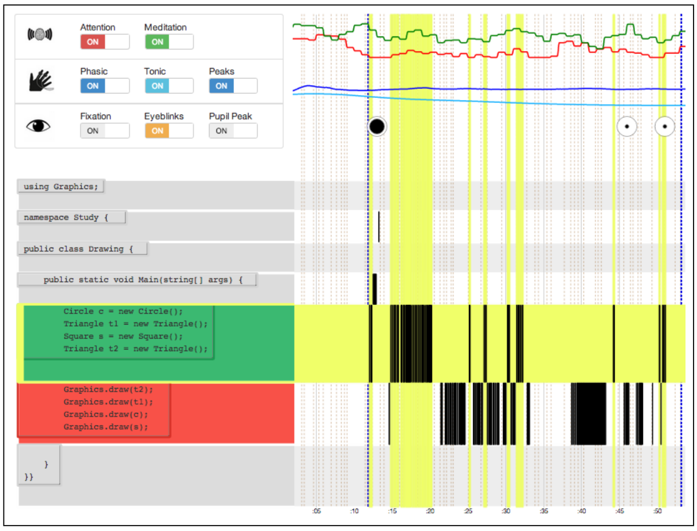
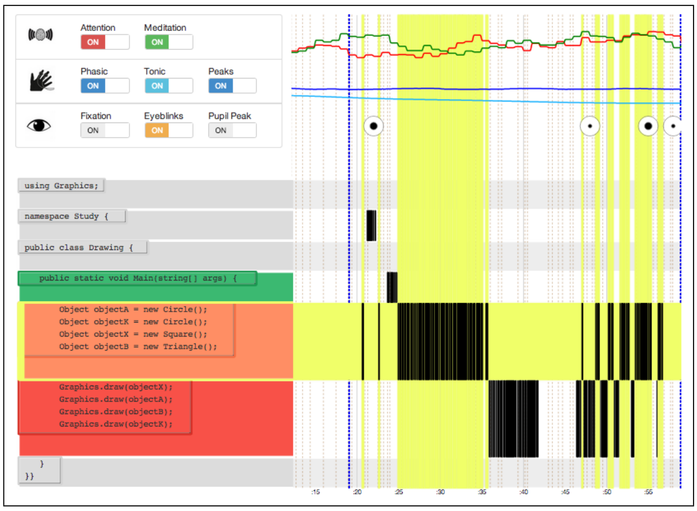


Figure 7.19: Task B2 (participant 13)

Task Duration: ***37s***
Fixations: ***112***
NASA TLX: ***4.3***
Diff. Ranking: ***1***
Correct: ***YES***

The time period where the participant focuses on the object initialization part is twice as long compared to task B1. In addition to that, more fixation hits in the AOI that contains the object initialization part are recognized in the last few seconds of the task.



[1] names that help to remember what the variable stands for

Analysis of AOI Measurements. For most of the participants, the comparison of fixation data between task B1 and task B2 show a similar picture as in the illustrated example in Figure 7.18 and 7.19. For 12 of 13 participants, there were clearly more (and also longer) fixation hits recognized in the object initialization segment of task B2 compared to the corresponding code block containing mnemonic variable definitions in task B1.

To compare pscho-physiological data that is measured while the participant focused on the object initialization, the *Grid View* can be used in addition to highlighting segments in the *Timeline View*. As already seen in the tool feature description, the visualization approach of the *Grid View* provides AOI-specific data. Extracts of the display output for task B1 and task B2 are given in Figure 7.20, whereas the columns show computed values for the following metrics: Δ*Median Attention*, Δ*Median Meditation*, *Mean Fixation Duration* (LTR). Because of the coloring scale which is computed by task, the background colors of the cells do not allow direct comparison between multiple tasks in relation to difficulty.

However, the cell data can be used to compare the object initialization areas. Comparing the values of the object initialization segment in the first column (Δ*Median Attention*), a higher value for task B1 can be noticed (13.54 to 7.38). Because this value represents the attention difference between the baseline and the task, it can be said that for the comparable AOI relatively lower attention values are measured during task B1 compared to task B2. The lower median attention value corresponds to the expected task differences regarding the concentration level. However, the results of a paired t-Test has shown that there was no significant difference; see box on the right. The same applies for the other two metrics shown in Figure 7.20.

> **T-Test (paired samples):**
> *H_0: AOI measurement values (ΔMedian Attention) of the comparable object initialization part in task B1 and in task B2 are the same, regardless of using mnemonic variables.*
>
> There was not a significant difference in the ΔMedian Attention for the object initialization AOI in task B1 (M=13.5, SD=17.5) and the object initialization AOI in task B2 (M=8.2, SD=14.4); $t(11)=1.182$, $p = 0.262$
>
> *Cannot Reject H_0*

Figure 7.20: Extracts of *Grid Views* for task B1 and task B2. Metrics in columns: Δ*Median Attention*, Δ*Median Meditation*, *Mean Fix. Duration* (LTR); all participants selected.

Analysis regarding Retrace Declaration Patterns. In a next step, an investigation regarding *retrace declaration patterns* is performed for the task pair under consideration. As mnemonic variables are used for easy remembering of a variables' meaning, an investigation whether there is a significant effect on the number of retrace actions seems to be appropriate. For that, the number of observed retrace declaration patterns is counted for each participant using the *Timeline View*.

The results are plotted in Figure 7.21. Based on this plot, a tendency to more pattern instances and a longer task duration can be identified for task B2 that makes use of generic variable names.
A t-Test was conducted to compare the effect of mnemonic variables on the number of retrace declaration pattern occurrences. The results of the test support the visual conclusion made based on Figure 7.21: There was a significant effect of using mnemonic variables (instead of generic ones) on the number of retrace declaration patterns at the p<.05 level; see box on the right.

T-Test (independent samples):
H_0: The number of retrace declaration pattern instances is the same regardless of using mnemonic variable names.

There was a significant difference in the number of retrace declaration patterns for the task B1 (mnemonic variable names) (M=4.07, SD=2.4) and task B2 (generic variable names) (M=7.4, SD=4.55); t(28)=-2.5, p = 0.018

Reject H_0

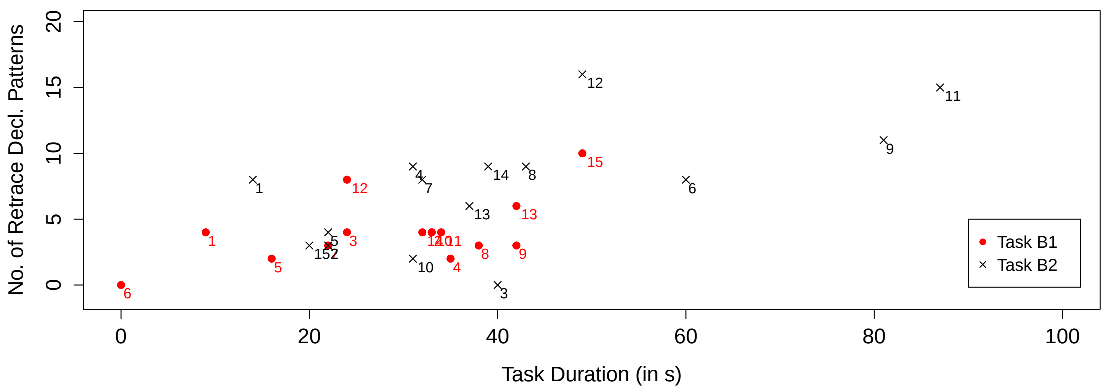


Figure 7.21: Number of *Retrace Declaration Patterns* for task pair B1-B2 (dot label specifies the participant).

Aggregated Metric Values. Table 7.3 provides mean metric values for the task pair B1-B2. It can be stated that the task duration in case of task B1 is lowered by 24%. Moreover, the task pair B1-B2 shows a noticeable difference regarding mean attention data, but both values are on a relatively low level. This can be explained by the short mean task duration of these tasks and the relatively low attention level that is often noticed in the first few seconds of a task.

Furthermore, the mean number of peak occurrences in the phasic EDA signal per time unit as well as the number of pupil diameter outliers per time unit is much higher for task B2. Although this difference seems to be significant, no significant difference was found. This implies that the metrics *EDA peaks per time unit* and *pupil diameter outliers per time unit* does not yield new results.

Task	False Answ.	Time (in s)	NASA TLX	Perc. Diff.	Fixations	Saccades	Fix./sec	Blinks/sec	Pupil P./min	EDA P./min	Attention	Meditation	ΔAttention	ΔMeditation
B1	0	31	4.2	1.5	95	127	3.1	0.60	10.5	1.4	39	53	11	8
B2	1	41	5.0	1.9	126	219	2.9	0.47	20.7	2.2	44	50	7	4

Table 7.3: Mean metric values over all the participants who completed task B1 and task B2 respectively.

Attention/Meditation Analysis. Based on the relatively high difference in the mean ΔAttention-value, further investigations are performed. For that, participant-specific data is plotted in Figure 7.22. The dot plot shows the metric values ΔMeditation and ΔAttention for each participant. Because the Δ-value is calculated by subtracting the mean value of the task phase from the mean value of the fish tank phase, a negative value indicates an increase in the particular metric. It can be stated that for 7 participants a larger attention increase can be identified for task B2 compared to task B1. Especially for participant P08 and P13 the difference in the ΔAttention-value between task B1 and B2 is quite high (>20). For 5 participants, task B1 provided a larger mean attention increase (no EEG data for P05/P07, P06 did not solve B1). In summary, although there exists a relatively high difference in the mean ΔAttention-value presented in Table 7.3, no clear trend can be found in the participant-by-participant consideration.

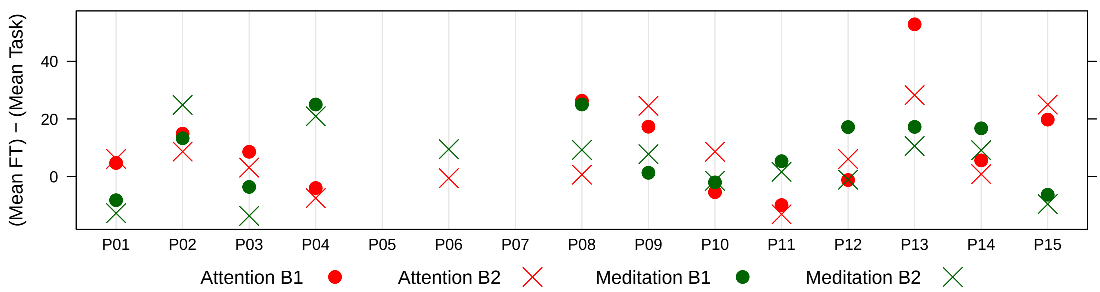


Figure 7.22: Dot plots representing norm. MindBand data (task pair B1-B2).

7.5.4 B1 vs. B3: Use of an Array

As for task B1, mnemonic variables are used in task B3 as well. But in task B, the shape objects are drawn by iterating over an array variable in which the shape objects are stored. It is expected that the participant has to switch multiple times between the object initialization area and the for loop to make sure in which order the shape objects are inserted into the array. Below, the corresponding timeline visualizations for a sample participant (P04) are shown. Again, the object initialization part is highlighted.

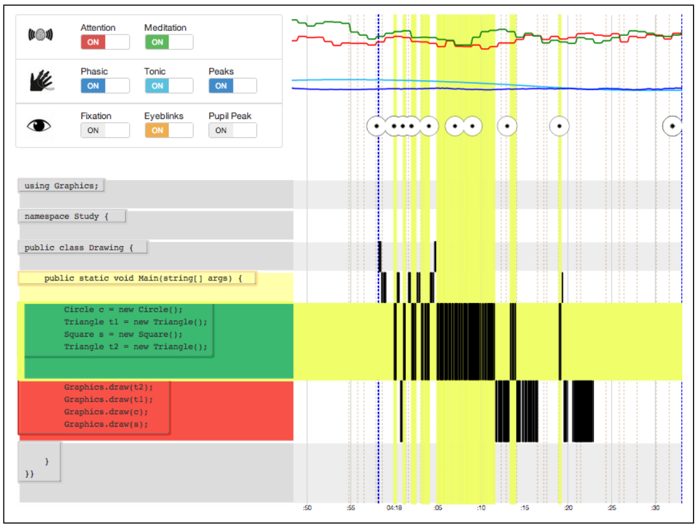


Figure 7.23: Task B1 (participant P04)

*Task Duration: **35s***
*Fixations: **98***
*NASA TLX: **4.13***
*Diff. Ranking: **2***
*Correct: **YES***

Attention level remains on a relatively low level over the complete task duration. A few pupil diameter outliers can be identified in the first 10s of the task. No EDA peak appears on the timeline.

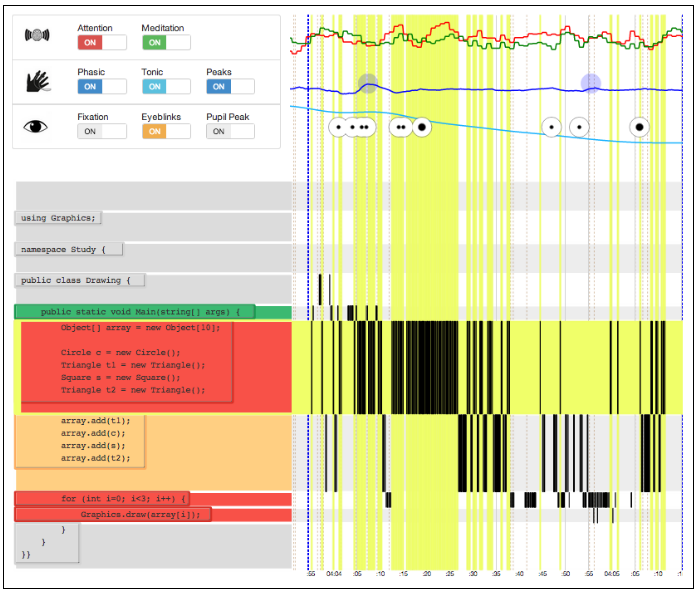


Figure 7.24: Task B3 (participant P04)

*Task Duration: **75s***
*Fixations: **194***
*NASA TLX: **5.53***
*Diff. Ranking: **3***
*Correct: **YES***

A high number of switches between the object initialization area, the array insertion and the for clause can be noticed. Also for this task, a few pupil diameter outliers can be observed at the beginning of the task. Additionally two EDA peaks appear on the timeline.

The *Timeline View* of other participants show a similar picture as in the illustrated example: The higher difficulty level of task B3 is reflected by a higher number of fixations and longer task completion time. Considering time normalized data for each participant, i.e. fixations per time unit, show interesting insights: A clearly lower fixation rate was observed for task B3 in comparison with task B1. The dot plot in Figure 7.25 illustrates this.

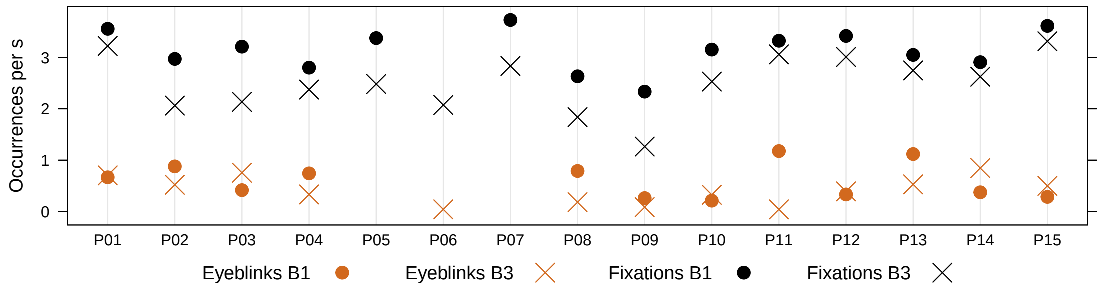


Figure 7.25: Dot plot representing fixation-/eyeblink occurrences per second (task pair B1-B3).

A lower fixation rate implies longer fixations. However, the *Grid View* can be used to retrieve AOI specific data, *e.g.*, the mean fixation duration. This could help to locate code segments with relatively long fixation durations. In Figure 7.26 extracts of the corresponding *Grid Views* are shown. In task B3 on the right hand side, relatively short fixations were recognized for the content of the loop statement (median: 195 ms). Relatively long fixations were measured for statements that add shape objects to the array (median: 331/329 ms). However, it is unlikely that these lines of code are the reason for the said difference in the fixation rate.

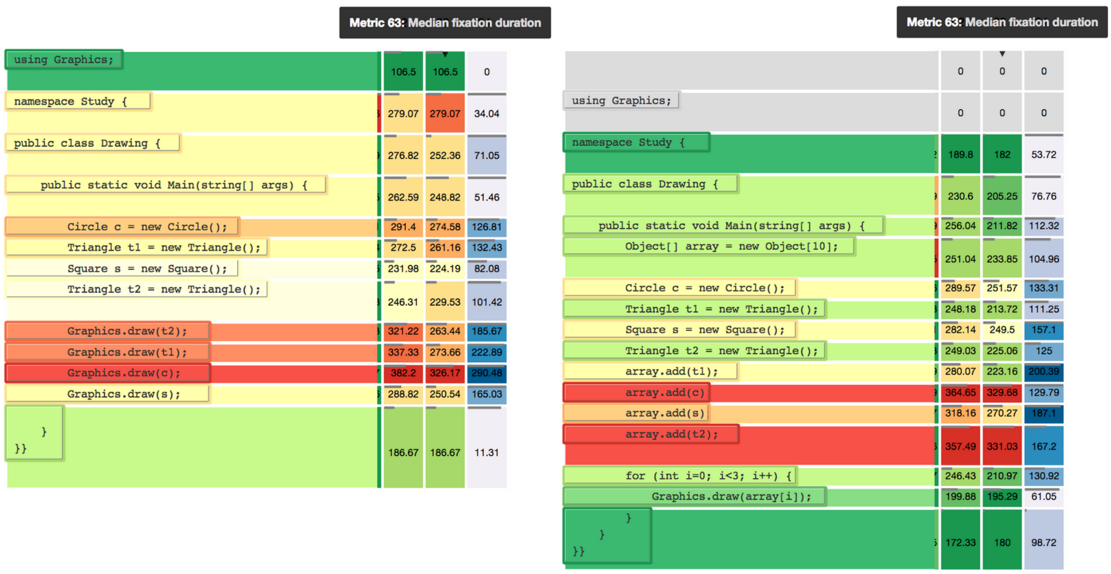


Figure 7.26: Extracts of *Grid Views* for task B1 and task B3. Metrics in columns: *Mean Fixation Duration*, *Median Fixation Duration*, *Std. Fixation Duration* (LTR); all participants are selected.

7.5.5 B3 vs. B6: Loop Complexity

Task B6 corresponds in its basic structure to task B3, but with the difference of a more complex loop drawing the shape objects on the screen. In Figure 7.27 and Figure 7.28, the corresponding visualizations with highlighted loop segments are shown for participant P02.

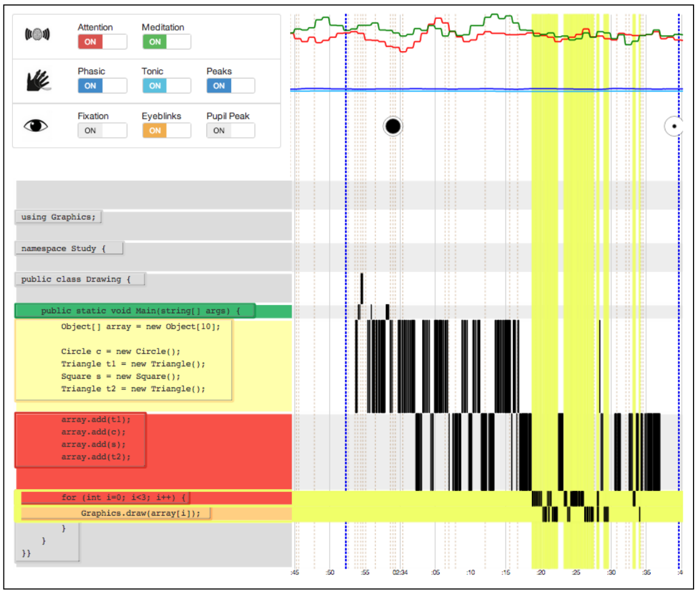


Figure 7.27: Task B3 (participant P02)

*Task Duration: **46s***
*Fixations: **122***
*NASA TLX: **6.13***
*Diff. Ranking: **3***
*Correct: **NO***

No peaks in phasic component of the EDA signal are recognized. An attention increase can be identified after about 10 seconds. A few seconds earlier, a sharp pupil diameter peak (>0.4mm above mean size) can be identified, indicated by the pupil icon on the timeline.

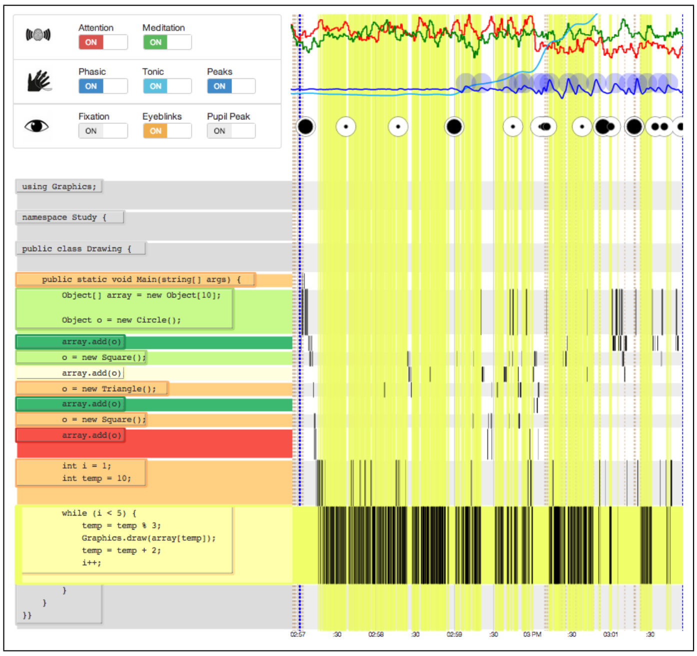


Figure 7.28: Task B6 (participant P02)

*Task Duration: **291s***
*Fixations: **600***
*NASA TLX: **5.73***
*Diff. Ranking: **6***
*Correct: **NO***

An accumulation of EDA peaks is recognized after about 2 minutes before the end of the task. After 3 minutes, a strong decrease of the attention level can be noticed. More than 80 percent of the fixations can be located in the while loop.

As the example for participant P02 in Figure 7.27 and Figure 7.28 already indicated, the participants took on average much more time to complete task B6 compared to task B3. Because of that, the mean number of fixations differ similarly. The corresponding numbers can be found in Table 7.4. In addition to that, a difference regarding the EDA peak occurrences (normalised by time) can be identified. On average, 1.3 EDA peaks were identified per minute for task B3, whereas the corresponding value for task B6 is more than 2.5 times greater.

Task	False Answ.	Time (in s)	NASA TLX	Perc. Diff.	Fixations	Saccades	Fix./sec	Blinks/sec	Pupil P./min	EDA P./min	Attention	Meditation	ΔAttention	ΔMeditation
B3	7	57	5.6	3.4	161	277	2.5	0.40	10.3	1.3	49	51	0	7
B6	9	231	10.1	7.0	554	891	2.8	0.34	5.4	3.4	46	51	7	7

Table 7.4: Mean metric values over all the participants who completed task B3 and task B6 respectively.

A more closer investigation regarding EDA peaks has shown that except of participant P09 for all participants more EDA peaks per time unit were observed for task B6 compared to B3; see Figure 7.29. However, it must be mentioned that the measure of electrodermal activity seems to be a measure that is strongly time-related. This means that the time that the participant already spent on the task has a strong influence on the measure. That suggests that occurrences of EDA peaks are not directly related to the difficulty of the task as such, although the difficulty of the task is closely related with the time the participant needs to complete the task.

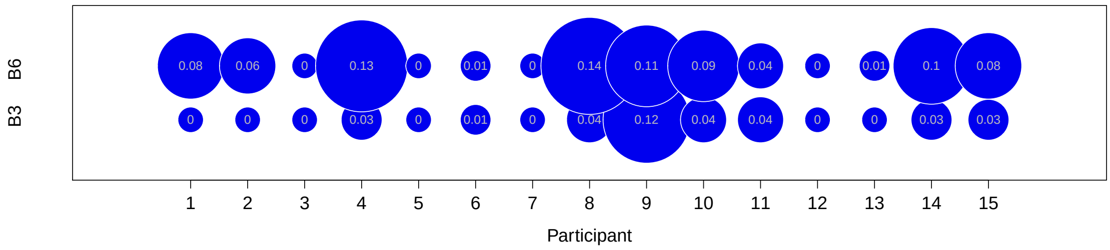


Figure 7.29: Bubble chart representing EDA peaks per second for task pair B3-B6.

7.5.6 B3 vs. B9: Additional Swap Method

In task B9, a swap method is defined that changes the position of the shape objects within the array. Multiple fixations between method call and method definition are expected. In case of participant P14 nearly 50 percent of the fixations are recognized in the swap method; see Figure 7.31.

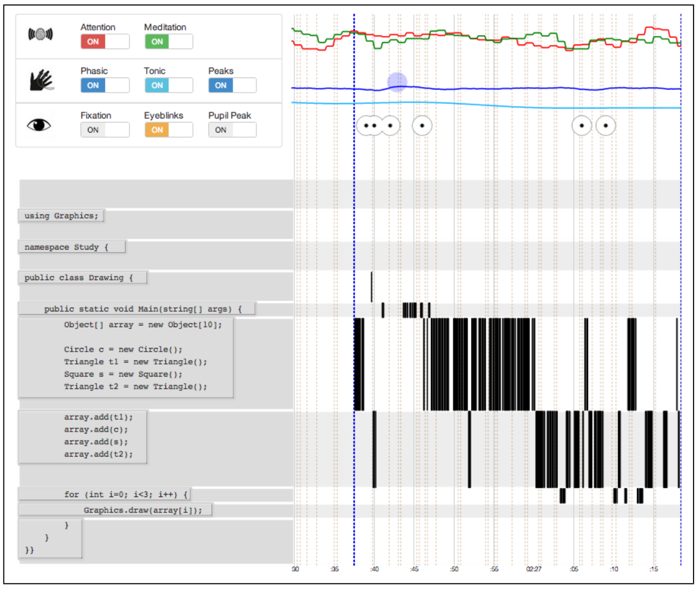


Figure 7.30: Task B3 (participant P14)

*Task Duration: **39s***
*Fixations: **104***
*NASA TLX: **7.0***
*Diff. Ranking: **3***
*Correct: **NO***

The participant took about 50 seconds for this task whereas only a few short fixations were recognized in the loop statement. Only one phasic EDA peak is noticed (about 7 seconds after the beginning of the task). Additionally, six pupil diameter peaks (>0.2mm above mean pupil size) can be observed.

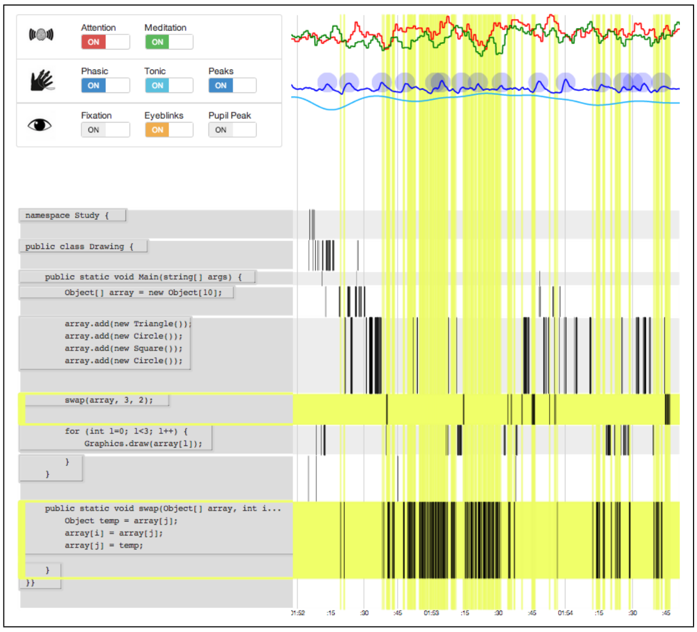


Figure 7.31: Task B9 (participant P14)

*Task Duration: **164s***
*Fixations: **475***
*NASA TLX: **14.4***
*Diff. Ranking: **7***
*Correct: **NO***

Nearly 50 percent of the fixations can be assigned to the swap method. The phasic EDA peaks are equally distributed over the task. The attention level at the end of the task (last 70s) is relatively high. No pupil diameter peaks appear on the timeline.

It is interesting to see that the number of fixations as well as the task duration is on average twice as long for task B9 than for task B3, but two participants completed task B9 with less than 100 fixations whereas for task B3 this is only one participant (Note, that nobody was able to solve task B9 correctly); see Figure 7.32. A high variance regarding the number of fixations can be recognized for task B9.

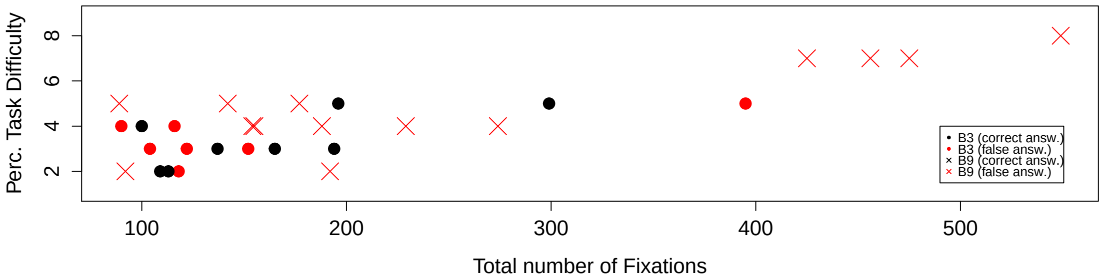


Figure 7.32: Scatter plot representing fixations and perceived difficulty (task pair B3-B9).

Furthermore, Table 7.5 shows that the mean attention value for task B9 is more than 10 percent lower than for task B3. The tendency of relatively low attention data for task B9 could be mentioned as an explanation for the low success rate of this task (14 wrong answers).

Task	False Answ.	Time (in s)	NASA TLX	Perc. Diff.	Fixations	Saccades	Fix./sec	Blinks/sec	Pupil P./min	EDA P./min	Attention	Meditation	ΔAttention	ΔMeditation
B3	7	57	5.6	3.4	161	277	2.5	0.40	10.3	1.3	49	51	0	7
B9	14	108	7.4	5.3	293	462	3.0	-	-	2.7	44	48	6	6

Table 7.5: Mean metric values over all the participants who completed task B3 and task B9 respectively.

7.5.7 B3 vs. B11: Easy Loop vs. Complex Object Init

In comparison to task B3, task B11 consists of a very complex object initialization line. The corresponding *Timeline Views* for participant P10 are shown below:

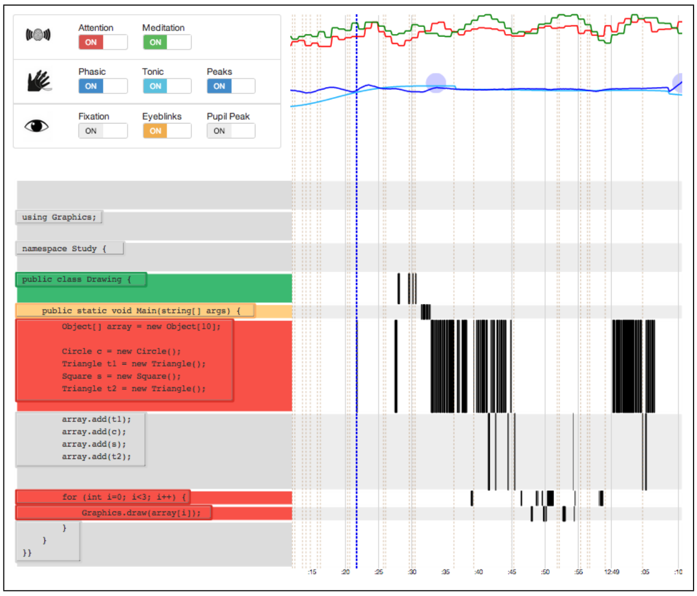


Figure 7.33: Task B3 (participant P10)

*Task Duration: **49s***
*Fixations: **137***
*NASA TLX: **2.1***
*Diff. Ranking: **2***
*Correct: **YES***

Most of the fixations can be assigned to the object initialization part. No pupil diameter outliers were observed and only two small EDA peaks appear on the timeline. Varying meditation data indicated by the wave form of the corresponding line chart is shown.

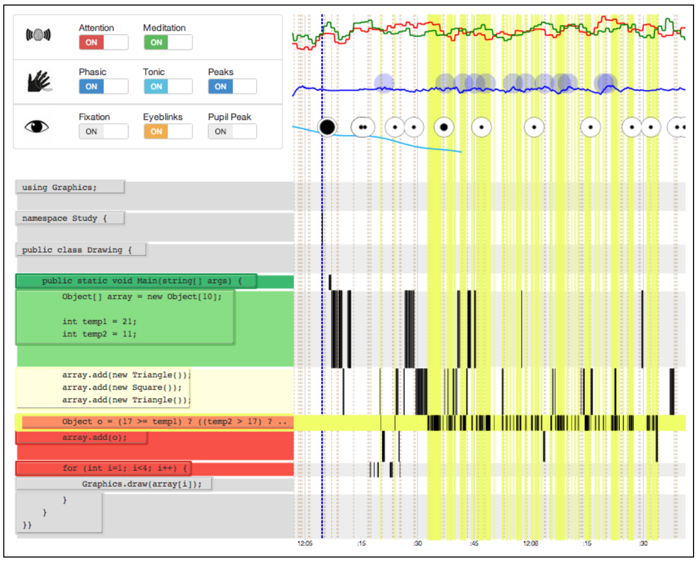


Figure 7.34: Task B11 (participant P10)

*Task Duration: **95s***
*Fixations: **291***
*NASA TLX: **7.0***
*Diff. Ranking: **5***
*Correct: **NO***

An accumulation of phasic EDA peaks is recognized (mainly while participant focused on the complex object initialization line). The pupil diameter peaks are more or less equally distributed over the complete task.

Table 7.6 provides some additional comparison data and gives an overview of task specific mean values over all the participants. Apart from the differences regarding the completion time and the perceived difficulty also disparities in psycho-physiological measures can be observed. For example the fixation rate seems to differ between the two tasks B3 and B11. Actually, for all 15 participants a higher fixation rate can be identified for task B11 in comparison to task B3. The low fixation rate in task B3 could be explained by the simple structure of the code and the use of mnemonic variables that prevents a high number of retrace fixations. This leads to a lower number of short fixations.

Task	False Answ.	Time (in s)	NASA TLX	Perc. Diff.	Fixations	Saccades	Fix./sec	Blinks/sec	Pupil P./min	EDA P./min	Attention	Meditation	ΔAttention	ΔMeditation
B3	7	57	5.6	3.4	161	277	2.5	0.40	10.3	1.3	49	51	0	7
B11	4	168	9.8	6.0	522	774	3.1	0.28	6.3	3.4	46	52	7	6

Table 7.6: Mean metric values over all the participants who completed task B3 and task B11 respectively.

Also the number of EDA peak occurrences per time unit is on average more than 2.5 times higher for task B11 (3.4 to 1.3 per min); see Table 7.6. A detailed overview of the number of EDA peaks per time unit for each participant is given in Figure 7.35. For 9 participants a higher number of EDA peaks per time unit appear on the timeline of task B11 compared to that of task B3.

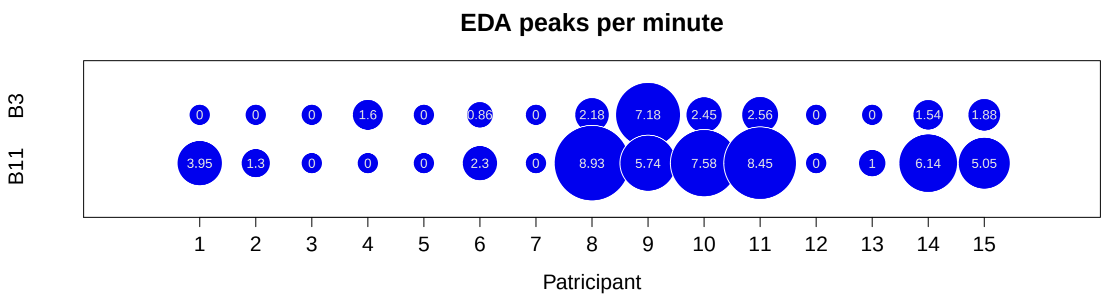


Figure 7.35: Bubble chart representing EDA peaks per time unit for task pair B3-B11.

7.5.8 B6 vs. B11: Moderate Loop vs. Complex Object Init

In Figure 7.36 and Figure 7.37, the timeline visualizations of participant P06 are depicted for task B6 and task B11. The loop statement is highlighted in both cases.

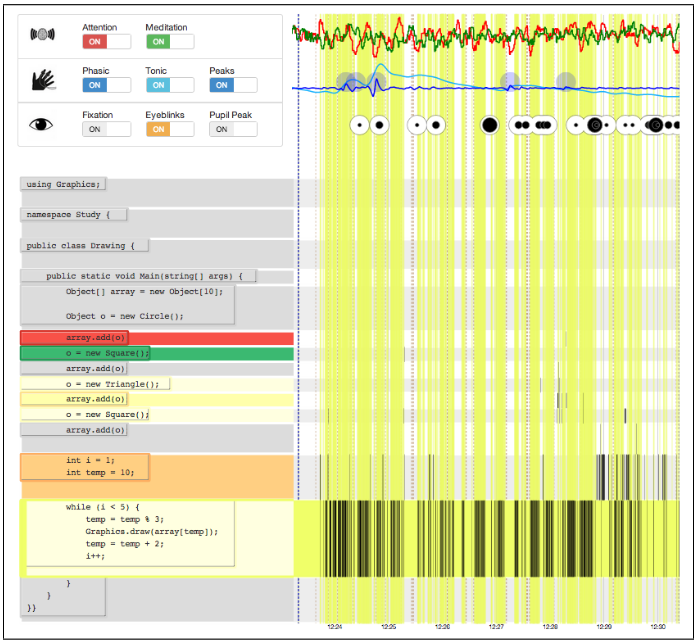


Figure 7.36: Task B6 (participant P06)

Task Duration: ***422s***
Fixations: ***875***
NASA TLX: ***13.2***
Diff. Ranking: ***7***
Correct: ***NO***

A tendency of a higher number of pupil diameter peaks can be recognized towards the end of the task. In addition to that, a strongly time-varying attention level can be identified. The fixations can be assigned to the loop statement in ∼90% of the time.

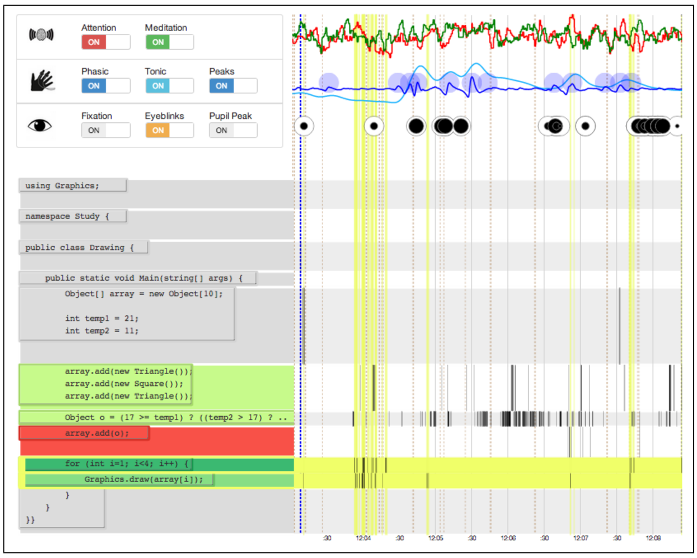


Figure 7.37: Task B11 (participant P06)

Task Duration: ***313s***
Fixations: ***573***
NASA TLX: ***10.1***
Diff. Ranking: ***6***
Correct: ***NO***

Decreasing attention values while participant focuses on the loop segment at the beginning of the task can be observed. As soon as the participant keeps the focus on the complex object initialization line, a few pupil diameter peaks are recognized as well as a number of phasic EDA peaks.

As Table 7.7 indicates, there are no significant differences between the two tasks regarding the psycho-physiological metrics.

Task	False Answ.	Time (in s)	NASA TLX	Perc. Diff.	Fixations	Saccades	Fix./sec	Blinks/sec	Pupil P./min	EDA P./min	Attention	Meditation	ΔAttention	ΔMeditation
B6	9	231	10.1	7.0	554	891	2.8	0.34	5.4	3.4	46	51	7	7
B11	4	168	9.8	6.0	522	774	3.1	0.28	6.3	3.4	46	52	7	6

Table 7.7: Mean metric values over all the participants who completed task B6 and task B11 respectively.

Differences regarding the fixation rate can be elaborated as the dot chart in Figure 7.38 illustrates. In 11 cases, the participant provided a higher fixation rate for task B11 compared to task B6. An analysis using the *Grid View* to locate code segments with relatively long fixations but . Because the loop statements of the tasks B3, B6 and B11 seems to be comparable AOIs, a investigation of AOI measurements using the Grid View was performed. However, no significant results were found. The results for the AOI metric ΔAttention are given in the box on the right.

ANOVA-Test:
H_0: *AOI measurement values* (i.e., Δ*Median Attention) of comparable loop statements in task B3, B6, B11 are the same regardless of the perceived task difficulty.*

There was a not a significant effect of loop complexity on the corresponding AOI measurement values (ΔMedian Attention) at the p<.05 level for the three tasks [F(2,32) = 2.51, p = 0.096].

Cannot Reject H_0

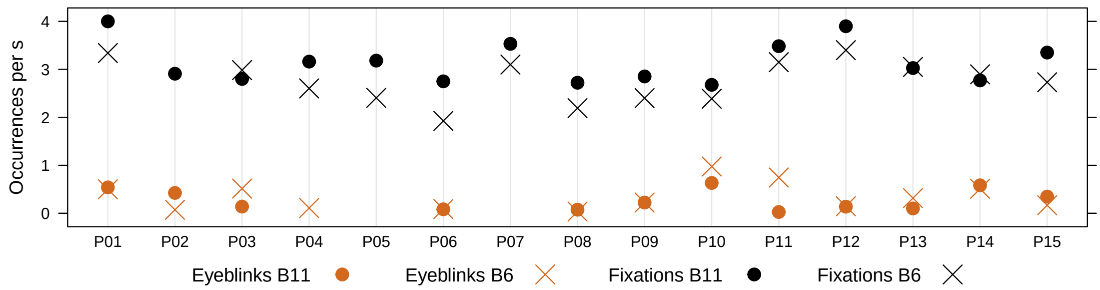


Figure 7.38: Dot plot representing fixation-/eyeblink occurrences per second (task pair B6-B11).

7.5.9 B9 vs. B11: Swap Method vs. Complex Object Init

Figure 7.39 and Figure 7.40 show the timeline visualizations of task B9 and B11 for participant P06. Again, the loop statement is highlighted in each task. Mainly due to the high number of fixations in the complex object initialization segment, this participant took about 4 times longer for task B11 than for B9.

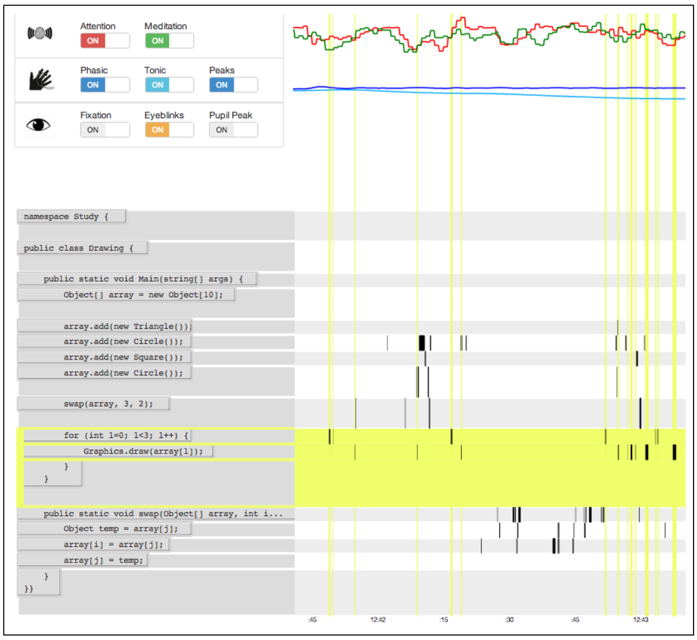


Figure 7.39: Task B9 (participant P06)

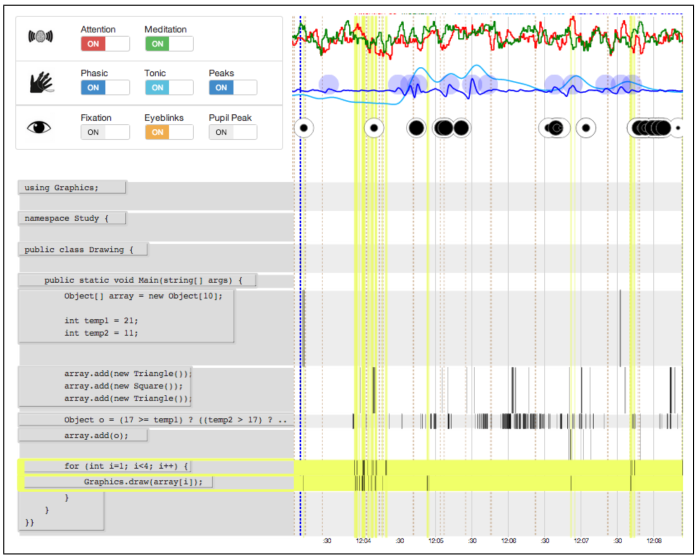
*Task Duration: **80s***
*Fixations: **154***
*NASA TLX: **4.4***
*Diff. Ranking: **4***
*Correct: **NO***

No pupil diameter peaks as well as phasic EDA peaks recognized during the complete task. The participant worked only 80 seconds on this code comprehension task and was not able to provide the correct answer (like all other participants).

Figure 7.40: Task B11 (participant P06)

*Task Duration: **313s***
*Fixations: **573***
*NASA TLX: **10.1***
*Diff. Ranking: **6***
*Correct: **NO***

A few pupil diameter peaks as well as a number of phasic EDA peaks are recognized for this task. Especially at the end of the tasks an accumulation of pupil diameter outliers (>0.4mm above mean pupil size) can be observed. In addition, a decrease in the attention is identified while the participant initially focuses on the loop statement for a longer time period.



Some comparison data for the task pair B9-B11 is given in the table below:

Task	False Answ.	Time (in s)	NASA TLX	Perc. Diff.	Fixations	Saccades	Fix./sec	Blinks/sec	Pupil P./min	EDA P./min	Attention	Meditation	ΔAttention	ΔMeditation
B9	14	108	7.4	5.3	293	462	3.0	-	-	2.7	44	48	6	6
B11	4	168	9.8	6.0	522	774	3.1	0.28	6.3	3.4	46	52	7	6

Table 7.8: Mean metric values over all the participants who completed task B9 and task B11 respectively.

The sample timeline visualizations in Figure 7.39 and Figure 7.40 indicate significant differences regarding the electrodermal activity (EDA). Time normalized EDA peak data is given for each participant in the bubble chart depicted in Figure 7.41. Based on the low number of participants that provide EDA peaks in this task pair, it is not possible to reliably tell how this metric differs in this task pair.

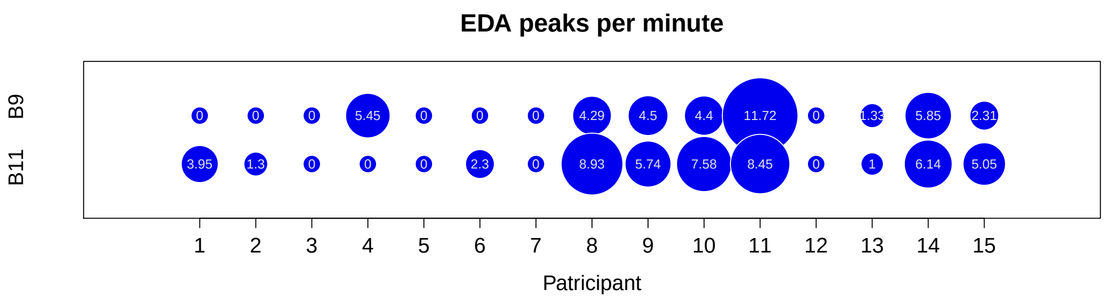


Figure 7.41: Bubble chart representing EDA peaks per time unit for task pair B3-B11.

7.6 Task-property specific Data Insights

In a further approach, all 116 individual tasks of the experimental study are categorized based on various properties. For example, the tasks are grouped into two different groups based on their NASA TLX score or the perceived difficulty ranking. The threshold of each metric is defined based on its median value, such that both groups consists more or less of the same amount of tasks. In addition to aggregated task metrics (*e.g.*, mean attention, mean attention), data is collected from the visualization outputs (*e.g.*, no. of pupil diameter peaks in the first 20 seconds of task time, first scan time, no. of scans, etc.). It is then elaborated what kind of differences between the two groups exist. The resulted mean values of this categorization approach are listed in Table 7.9. Note that tasks with missing/trunked data are omitted. Table 7.9 provides the following insights:

- The duration of the first scan is is twice as long for the group with tasks that are ranked as more difficult. The results of an independent t-Test for the groups *task difficulty ranked* > *4* and *task difficulty ranked* $\leq$ *4* show that there is a significant difference; see 1st box on the right.
- A slightly higher mean attention value can be recognized for the tasks that were not solved correctly compared to the ones that are solved correctly; see 2nd box on the right.
- The average number of EDA peaks in the first 10/20/60 seconds of a task is considerably higher for tasks that are ranked as more difficult; see 3th box on the right.
- Mean attention for tasks that are rated as more difficult (*difficulty rating* > *4*) is not higher than for tasks that are ranked easier relatively to the rest.
- No significant difference between number of pupil diameter peaks for the two difficulty level groups is recognized.
- No participant performed more than one scan for a task of type B.
- No significant difference in mean attention between long tasks (> 60s) and short tasks ($\leq$ 60s).
- No significant difference in the number of pupil diameter peaks in the first 60s for task that are perceived as relatively easy (*perceived difficulty rating* $\leq$ *4*) and tasks that are perceived as relatively difficult (*perceived difficulty rating* > *4*).

T-Test (independent samples):
H_0: *The first scan time is the same regardless whether the participants perceives the task as relatively easy (value* $\leq$ *4) or relatively difficult (value > 4).*

There was a significant difference in the first scan time for the group of tasks that are perceived as relatively easy (value $\leq$ 4) (M=14.6, SD=10.0) and the group of tasks that are perceived as relatively difficult (value > 4) (M=30.6, SD=20.4); t(89)=13.84, p < 0.001.

Reject H_0

T-Test (independent samples):
H_0: *The mean attention of a task is the same regardless whether the task is solved correctly or not.*

There was not a significant difference in the for the group of tasks that were solved correctly (M=44.2, SD=11.92) and the group of tasks that are were not solved correctly (M=46.3, SD=9.18); t(89)=0.283, p = 0.77.

Cannot Reject H_0

T-Test (independent samples):
H_0: *The number of EDA peak occurrences in the start phase of a task (first 20s) is the same regardless of participants perceived difficulty ranking (Note that duration of some of the tasks is less than 20s).*

There was not a significant difference in number of EDA peaks for tasks that are perceived as relatively easy (value $\leq$ 4) (M_{20s}=0.68, SD_{20s}=0.85) and for tasks that are perceived as relatively difficult (value > 4) (M_{20s}=1.17, SD_{20s}=1.2); p_{10s} = 0.033, p_{20s} = 0.051.

Cannot Reject H_0

	Kind of Task		NASA TLX		Difficulty		Duration		Correctness	
Metric	A	B	< 6.0	> 6.0	< 4	> 4	< 60 s	> 60 s	NO	YES
NASA TLX	8.9	6.0	3.7	9.8	4.8	9.6	5.0	8.5	8.2	6.4
Duration	124.2	86.2	65.0	127.9	46.5	165.7	39.2	149.9	133.4	86.3
Attention	44.8	44.6	44.3	45.0	45.1	44.1	44.4	44.9	46.3	44.2
Meditation	51.1	50.2	51.5	49.4	51.4	49.2	52.1	48.9	49.8	50.6
Fixations	289.2	239.4	258.9	248.0	248.7	259.8	211.8	291.6	297.6	240.5
Saccades	411.1	391.6	465.0	331.0	394.7	400.3	346.8	443.3	453.1	380.8
Scans	2.5	1.0	1.1	1.8	1.2	1.8	1.2	1.6	1.3	1.5
1st Scan Time	20.7	21.7	16.7	25.9	14.6	30.6	13.4	28.7	23.8	20.7
EP in first 10 s	0.4	0.5	0.5	0.4	0.3	0.6	0.2	0.6	0.4	0.4
EP in first 20 s	0.6	1.0	1.1	0.7	0.7	1.2	0.6	1.2	0.9	0.9
EP in first 60 s	1.8	1.9	1.8	1.9	1.2	2.8	1.0	2.7	1.8	1.9
Total EP	4.4	4.1	3.2	5.2	1.3	8.1	1.2	7.0	5.4	3.8
PP in first 10 s	1.3	1.9	2.0	1.4	1.9	1.5	1.8	1.7	1.4	1.8
PP in first 20 s	3.0	3.0	3.6	2.4	3.1	2.9	3.0	3.0	2.3	3.2
PP in first 60 s	6.8	4.6	5.8	4.6	5.0	5.5	4.4	6.0	4.3	5.5
Total PP	11.1	7.5	8.0	9.0	5.9	12.0	4.9	11.8	10.1	8.0
EP / min	1.1	1.6	1.9	1.1	1.5	1.4	1.5	1.4	1.2	1.6
PP / min	5.3	5.7	7.7	3.5	7.6	2.9	7.8	3.5	3.6	6.2

Table 7.9: Mean metric values for task-property groups (EP = EDA peaks, PP = pupil diameter peaks).

7.7 Metric Dependency Analysis

Another analysis approach focuses on correlations between multiple measurements within task data. For that, the *Timeline View* is used. A sample question would be: Does the attention level correlates with pupil peak occurrences? In this case, the *Timeline View* is used to identify in how many cases that a high attention level goes with pupil diameter outliers. In a first step, the analysis is done by visual inspection of single tasks. If a probable correlation is identified, an evaluation is conducted. In the following, the results are summarized:

Eyeblinks - Attention Level. Considering eyeblink data (displayed as dashed lines in the *Timeline View*), it was found that in a number of cases an accumulation of eyeblinks correlates with a relatively low attention level. Figure 7.42 illustrates that. To verify this finding, a time series analysis would be needed which is beyond the scope of this thesis.

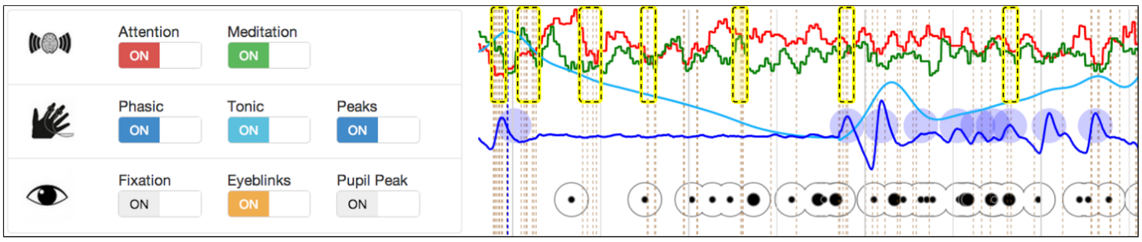


Figure 7.42: Probable correlation between attention and eyeblinks (task B11, participant P06).

Eyeblinks - EDA Peaks. In related work, it was found that the eyeblink rate can be used as an indicator for arousal [DJM90]. The same applies for the measure of electrodermal activity [Bou12, DSF07]. In some of the timeline visualizations it can be found, that an accumulation of eyeblinks goes with a peak in the phasic EDA signal; see Figure 7.42. However, a Pearson correlation coefficient was computed to assess the relationship between the fixation rate and the number of EDA peak occurrences per time unit. The results show that there was no significant correlation; see box on the right.

Pearson's Test:
H_0: *Eyeblink rate is regardless of the number of EDA peak occurrences per time unit.*

There was no correlation between the two variables eyeblink rate and number of EDA peaks per time unit [r = -0.08, n = 114, p = 0.391].

Cannot Reject H_0

Fixation Rate - Eyeblink Rate. A participant-specific data inspection has indicated a correlation between fixation rate and eyeblink rate. Eyeblink rate and fixation rate are plotted in Figure 7.43. As the scatter plot illustrates, the data for participant P01 show clear correlation tendencies whereas for participant P06 the measures under consideration do not correlate at all. However, a Pearson's test that was conducted on the complete dataset show low presumption against the null hypothesis; see box on the right.

T-Test (independent samples):
H_0: *The number of eyeblink occurrences per time unit is the same regardless how many fixations occurred.*

There was no correlation between the two variables fixation rate of a task and eyeblink rate of task [r = 0.208, n = 85, p = 0.053].

Cannot Reject H_0

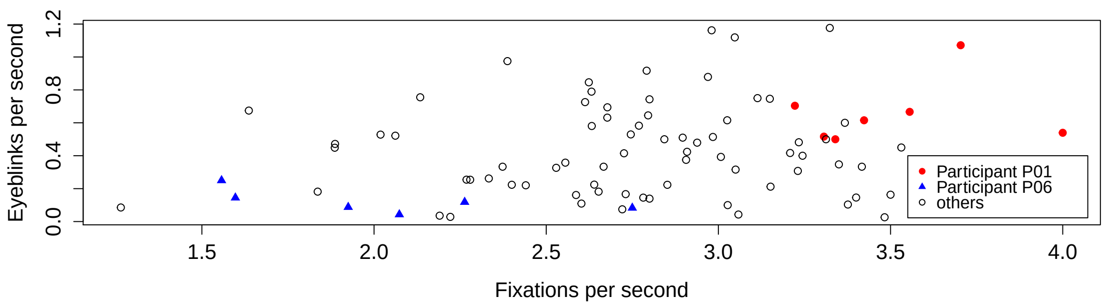


Figure 7.43: Scatterplot representing fixation rate and eyeblink rate for each of the task.

EDA Phasic Peaks - Attention Level. It was found that in some cases EDA phasic peaks were recognized on the timeline where a local minimum of attention is noticed. Figure 7.44 shows an example where in four cases the occurrence of an EDA peak correlates with a relatively low attention value, followed by an attention increase (illustrated by arrows).

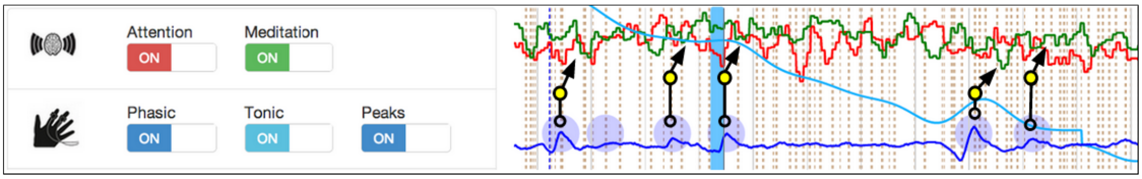


Figure 7.44: Probable correlation between EDA and attention data (task B6, participant P11).

An investigation whether the described correlation can be verified is performed over some other tasks. The highest percentage of EDA peaks that go with a relatively low attention level was found for participant P15: On average for 85% of all the EDA peak of a task, low attention can be observed (Note that task A2, B1, B6, B11 were considered only based on number of EDA peaks). For the other participants considerably lower values were computed. As participant P11 (task B6 depicted in Figure 7.44) provides only for two tasks EDA data with an appropriate number of peaks, it could not be proven as a participant specific correlation pattern.

Pupil Diameter Peaks - EDA Peaks. The measure of electrodermal activity can be used as an indicator for arousal or fear, whereas pupil dilation indicates high cognitive load [Ax53,BI08]. Nevertheless, in some cases, as for example for the task given in Figure 7.45, slightly time-shifted correlation tendencies can be identified. However, there is no significant amount of similar tendencies. A further investigation was performed to prove the correlation between these to measures in the start phase of a task: A correlation test was conducted to prove whether there was a correlation between the number of pupil diameter outliers (after 10/20s) and the number of EDA peak occurrences (after 10s/20s). The results of this test are shown in the box on the right.

Pearson's Test):
H_0: The number of phasic signal peak occurrences is the same regardless how many pupil diameter peaks that can be noticed in the same time period (No. of peaks after 10s/20s; Note that duration of some tasks is less than 20s).

There was no correlation between the two variables [$r_{10s} = 0.152$, $n = 70$, $p_{10s} = 0.202$].
There was no correlation between the two variables [$r_{20s} = 0.101$, $n = 70$, $p_{20s} = 0.400$].

Cannot Reject H_0

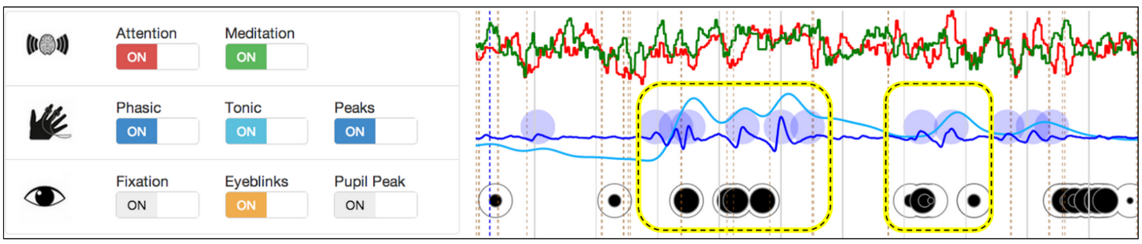


Figure 7.45: EDA phasic peak and pupil diameter peaks on timeline (task B11, participant P06).

7.8 Summary

Below, statistical test results are summarized. Significant results are underlined.

Category	Description and Null Hypothesis	Sign. Test	Result
Eye movements	*First Scan Time - Task Performance* H_0: Task performance (correct/false answer) is regardless of scan time (Note: performed for task B3 and B11 only)	*T-Test:* (For Tasks B3/B11) t(12) = 0.195/0.081 p = 0.848/0.937	Cannot Reject H_0
	<u>*First Scan Time - Perceived Difficulty* H_0: Scan time is regardless of perceived difficulty ranking.</u>	*Pearson's Test:* r = 0.393, n = 106, p = < 0.001	Reject H_0
	Retrace Declaration/Reference Pattern see "Effect of Mnemonic Variables".		
	<u>*Multiple Scan Pattern* H_0: The number of scans is regardless of the task performance (correct/false, considering tasks A1/A2 only).</u>	*T-Test:* t(27) = 2.908 p = 0.007	Reject H_0
Comparable Task Pair Analysis	<u>*Effect of Mnemonic Variables (B1-B2)* H_0: The number of retrace declaration pattern instances is the same regardless of using mnemonic variable names.</u>	*T-Test:* t(28)=-2.503 p=0.0184	Reject H_0
	AOI Measurements (B1-B2) H_0: AOI measurement values (*i.e.*, ΔMedian Attention) of the comparable object initialization part in task B1 and B2 are the same.	*Paired T-Test:* t(11) = 1.182 p = 0.262	Cannot Reject H_0
	AOI Measurements (B3-B6-B11) H_0: AOI measurement values (*i.e.*, ΔMedian Attention) of comparable loop statements in task B3, B6, B11 are the same regardless of the perceived task difficulty.	*ANOVA Test:* F(2,32)=2.51 p=0.096	Cannot Reject H_0 B_{11}: M=9.318 B_3: M=-6.682 B_6: M=8.615
	<u>*Order of Field Assignments (A1-A2)* H_0: The number of fixation is the same regardless whether the field assignments follow a logical order or not.</u>	*Paired T-Test:* t(13) = -2.634 p = 0.021	Reject H_0

Time-related Insights	*Rising Attention* No test was conducted (Requires time series analysis which would be beyond the scope of this thesis).		
	Electrodermal Activity & Pupillometry No test was conducted (Requires time series analysis).		
	Pupillometry H_0: The number of pupil diameter peak occurrences in the start phase of a task (first 20s) is the same regardless of participants perceived difficulty.	*Pearson's Test:* r = -0.089, n = 71, p = 0.463	Cannot Reject H_0
Task-property specific Insights	*Difficulty Rating vs. First Scan Time* H_0: The first scan time is the same regardless of participants perceived difficulty ranking.	*T-Test:* F(1,89) = 13.840 p = < 0.001	Reject H_0
	Difficulty Rating vs. No. of EDA peaks H_0: The number of EDA peak occurrences in the start phase of a task (first 10/20/60s) is the same regardless of participants perceived difficulty ranking (Note that duration of some of the tasks is less than 60s).	*T-Test:* (after 10s/20s/60s) F(1,89) = 4.7/3.9/11.3 p_{10s} = 0.033 p_{20s} = 0.051 p_{60s} = 0.001	Cannot Reject H_0
Metric Dependency Insights	*Fixation Rate - Eyeblink Rate* H_0: The number of eyeblink occurrences per time unit is the same regardless how many fixations occurred.	*Pearson's Test:* r = 0.208 n = 85 p = 0.053	Cannot Reject H_0
	Pupil Diameter Peaks - EDA peaks H_0: The number of phasic signal peak occurrences is the same regardless how many pupil diameter peaks that can be noticed in the same time period (No. of peaks after 10s/20s/60s; Note that duration of some tasks is less than 60s).	*Pearson's Test:* (after 10s/20s/60s) r = 0.152/0.101/0.013 n = 70 p = 0.202/0.400/0.913	Cannot Reject H_0
	Eye blinks - Attention No test was conducted (Requires time series analysis).		

Table 7.10: Overview of statistical test results.

Chapter 8

Recap

In this chapter, the contributions of the thesis are summarized and the research questions are answered. Additionally, limitations concerning the data visualization approaches and the computed metrics as well as aspects of future work are given.

8.1 Summary of Contributions

This thesis provides the following contributions:

- *Definition of a visualization concept to inspect psycho-physiological data recorded from developers working on code comprehension tasks:* A design and interaction concept for the visualization prototype is elaborated. A set of potential visualization approaches that could support the pattern identification process is given.
- *Providing AOI (area of interest) measurement computations to classify specific parts of the code:* Based on the eye-tracking data, measured psycho-pyhsiological data can be assigned to specific code segments/areas of interest (AOIs). Matlab scripts are provided to compute AOI-specific metric values. In total, 45 metrics were defined. The computation of each metric is performed for all the segments of all the tasks, participant by participant. Finally, the computed AOI measurements are used for the visualization approach.
- *Implementation of a JavaScript web application that visualizes psycho-physiological data along with the corresponding code analysis tasks:* The implemented visualization prototype consists of three different views whereas each focuses on a different data analysis technique. Using the highlighting feature, psycho-physiological data can be related to specific lines of code. In addition to that, code segments can be selected to mark up the EDA, EEG and pupillometry data that was measured while the participant has focused on that code segment. Apart from the *Timeline View*, two other views named as *Grid View* and *Aggregated Timeline View* were implemented. Using these views, aggregated data for multiple participants can be visualized.
- *Conducting a visual analysis of EDA, EEG and eye-tracking data that was recorded for eight different code comprehension tasks of varying difficulty level:* For all the 116 individual data recordings in the given dataset, the output visualizations are analyzed. Various analysis approaches are used to discover insights. This includes analyses that focuses on the identification of code reading patterns, the elaboration of differences in comparable task pairs, the identification of metric correlation within tasks as well as the detection of task-, time- and participant-related patterns.

- *Evaluating the findings of visual inspection by performing statistical significance tests:* Based on the findings of the visual inspection, statistical significance tests were conducted. Below, the patterns that were verified by significant results are listed:
  - It was found that there is a *correlation between the first scan time and the perceived difficulty* specified by the participant (*first scan time* = initial reading of the code until 80% of the lines are reached). The participants took significantly longer for the first scan if the task is perceived as more difficult.
  - Comparing the fixation data of correctly solved tasks with the fixation data of tasks where the participant was not able to find the correct answer has shown that there is a relationship between the number of scans and the task performance. It was found that in fixation data that is related to tasks that were not solved correctly a significant higher number of scans can be observed. This pattern is named in the work at hand as *Multiple Scan Pattern* and can be interpreted as a cognitive action of reading the code multiple times without recognizing the crucial parts of the task.
  - In the task pair analysis, it was found that the *use of mnemonic variable names leads to a significant lower number of retrace declaration pattern occurrences*.
  - Based on task pair A1-A2 it was found that order of the field assignments has a signficant effect on the total number of fixations: If the field assignment are ordered in such a way that they follow a logical order instead of a random order, a *significantly lower number of fixations is required* to complete the task.

8.2 Conclusions

The purpose of this thesis was to discover patterns in psycho-physiological data that was recorded while software developers were working on code comprehension tasks. The dataset used in this project consists of eye-tracking-, EEG- and EDA data. Additionally, general task information data as for example the task duration or the perceived difficulty is provided. Although multiple visualization approaches were elaborated, only one of them has proven to be suitable for a detailed analysis of the given dataset. In the following, the research questions that focus on the visualization approaches (*RQ1*), the pattern finding process (*RQ2*) and the use of code segment specific data (*RQ3*) are revised and reflected upon:

(*RQ1*) *How can psycho-physiological measurements be visualized to support the analysis of patterns in the data that can be related to difficulties a software developer might have experienced while working on a specific part of the code?*

Various approaches that visualize biometric data along with the corresponding code comprehension task are presented in this work. It was found, that the *Timeline View* is best suited for a detailed data analysis, but does not allow to show data for multiple participants at once. To compare multiple participants with each other, storing the visualization outputs as images is required. Using the highlighting feature that is integrated into the *Timeline View*, psycho-physiological data can be related to specific lines of code. Additionally, the prototype allows for the detection of difficult code segments based on psycho-physiological measurements (*e.g.*, indicated by peaks in the phasic component of the EDA signal). Also aggregated data visualization approaches were taken into account. However, the approaches named *Grid View* and *Aggregated Timeline View* did not seem to be useful for a detailed data analysis.

(*RQ2*) *Which reoccurring patterns within code comprehension tasks can be identified in psycho-physiological data using visual inspection?*

Code reading patterns as for example the *scan pattern*, the *retrace declaration pattern* as well as the *retrace declaration pattern* that are all defined by Uwano *et al.* were also found in the dataset used in this project [UNMM06]. An analysis that aims at investigating the effect of using mnemonic variable names in relation to the number of retrace declaration patterns has shown a significant result. Additionally, correlation between the *first scan time* and the perceived difficulty of a task was found. Moreover, a relationship between the task performance and the number of scans was observed (*Multiple Scan Pattern*). Apart from the code reading patterns, findings related to psycho-phyisological measures such as electrodermal activity or the EEG signal were found (*e.g.*, time related findings: rising attention level, sharp EDA peaks in the last phase of a task). Based on the findings of the visual inspection, a set of patterns was suggested. Although some of the findings seemed to be promising, no significant test results were found. Because of that, it is misleading to speak in these cases in terms of patterns.

(*RQ3*) *Given a pair of tasks that have been specifically designed to provide two different difficulty levels, can we explain the difference in difficulty by the AOI measurements?*

The table like approach named *Grid View* presents several AOI metrics which allow to inspect psycho-physiological data that can be related to specific code segments. However, it was found that the values presented in the *Grid View*, called AOI measurements, should be treated with caution. Only the metrics related to eye-tracking data have shown reliable information (*e.g.*, *Mean Fixation Duration within AOI*). Nevertheless, an analysis for comparable code segments was performed. The results have indicated that there is no significant difference. Based on the low number of comparable areas of interest that are of large enough size (to ensure a reliable number of fixation hits), the use of AOI measurements to explain the difference in the difficulty for task pairs cannot be evaluated in more detail with the given dataset.

8.3 Limitations and Threats to Validity

Limitations of the visualization approaches. Some key limitations of the implemented visualization approaches are summarized below:

- The aggregated data visualization approach *Aggregated Timeline View* lacks of detailed analysis. The fact that the task completion time strongly varies between the participants makes meaningful aggregated data analysis for multiple participants difficult. This is especially true for the identification of an aggregated fixation path. Although the *Aggregated Timeline View* includes such an approach in form of a heat map like visualization, it cannot be used to identify patterns that can be related *e.g.*, to the mean number of scans or the *first scan time*.
- Also the *Grid View* that was designed to analyze code segment specific measurements seems not to be a suitable approach. Especially for areas of interest (AOIs) where only a few hits were recognized, the computed metric values that correspond to specific code segments are not representative. This applies especially to small code segments (single lines of code). It was shown that the variance of measurements that are related to specific AOIs is in most of the cases quite high.
- The *Timeline View* does not allow the comparison of multiple participants at once. Comparing the display outputs of multiple visualization requests is required instead. In addition to

that, it is quite inconvenient to identify metric information such as the *first scan time* or the number of *retrace patterns* from the individual *Timeline Views*.

Data Limitations. Limitations concerning the used data and the computed metrics are listed below:

- Out of the 15 participants, no EDA data is provided for three participants (P05, P07 and P12). In addition to that, a few participants provided rarely EDA peaks. To make reliable statements how electrodermal activity can be related with the difficulty of specific code segments requires much more data.
- For the metric representing the number of pupil diameter peaks per time unit, participant-related differences were found (*e.g.*, P11 provides for three tasks extraordinary large values) which actually strongly influences the mean value. However, task-specific findings related to time normalized pupil diameter outliers fail to appear.

External Validity. In this section, it is described to which extent the results of the pattern finding analysis can be held to be valid for other code comprehension tasks. All the code comprehension tasks for the given dataset are quite short and contain a maximum of 29 lines of code. To prove whether the findings of this thesis can be held to be valid for longer tasks, additional lab experiments have to be conducted. Since all the participants that took part in the lab experiment were professional software developers it can be assumed that the findings can be generalized to other people that have to deal with code comprehension activities.

Internal Validity. In internal validity, the main question for the thesis is whether the findings (*e.g.*, effect of mnemonic variables on the number of retrace declaration pattern occurrences) are based on the difference in the task difficulty and not because of other possible causes. Due to the similarity of some of the tasks, a learning effect is possible. To counter-act this, the task order for each participant is randomized. Additionally, the task order could also have an influence on the measured psycho-physiological data. Although there is a break of two minutes between each of the tasks (mind relaxation phase), influences due to high mental load in preceding tasks are possible and could lead to concentration difficulties. This is especially true for the later tasks in the task order (total duration to complete the experiment: 1.5 h). Since the experiment has been conducted in a lab environment, it is possible that the participants may have performed differently than at their own work place.

Construct Validity. A threat to the experimental study is that there exist apart from the difficulty of a task or the complexity of certain parts of the code other factors that might influence psycho-physiological measures (*e.g.*, stress, personality, health status).

8.4 Future Work

More research in this field is needed for a better understanding of psycho-physiological measures in relation with a software developers' coding activities. Begel, co-author of the work this thesis is based on, said recently: "We are still at the experimental stage, learning to understand what all these sensors are telling us about the software developers" [Cur14]. To learn how software developers read software code and how they react in specific parts of coding activities, considerably more work will need to be done.

Further research in that field with special focus on code comprehension activities might explore the potential of the presented visualization approach for other code comprehension studies that make use of psycho-physiological measures. A future study with the same experimental set up but with considerably longer tasks would be very interesting. In addition to that, studies that make use of code comprehension tasks in other programming languages could asses whether there are any language-related differences (*e.g.*, regarding code reading patterns). The investigation in this thesis has shown that an even higher number of study subjects is required to reliably discover patterns. It was shown that psycho-physiological sensor data differ clearly between some of the participants. It would be interesting to compare groups of participants with similar data. However, in the study this thesis is based on, the number of participants is too small to reliably tell how such groups differ.

It was shown that the implemented visualization solution has some limitations regarding the comparison of data from multiple participants. The *Timeline View* currently does not support displaying multiple timelines at once. In the thesis at hand, screenshots are captured for each single timeline visualization output to guarantee a systematic analysis. In studies with high number of participants such an analysis approach would be inconvenient. However, an approach that allows to compare psycho-physiological data over multiple participants is desirable to discover patterns more easily. A possible solution to this problem would be an additional feature that allows to arrange multiple timeline visualization outputs in a table-like structure. To facilitate the data analysis further, supporting features that automatically calculate and visually highlight pattern related measures is desired. For instance, an algorithm that automatically computes the *first scan time* could be implemented. Moreover, the number of *retrace declaration pattern* occurrences could be computed by considering fixation path data. Based on the resulting fixation related measures (*i.e.*, scans, retrace declaration/reference patterns) highlighting on the timeline visualization could appear accordingly to improve the data analysis process further.

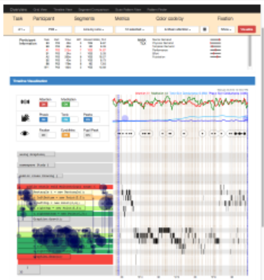
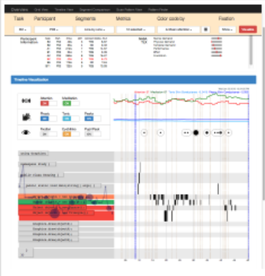
However, considerably more work will need to be done to reliably determine patterns using psycho-physiological sensors that can be used to develop novel programming support tools.

Appendices

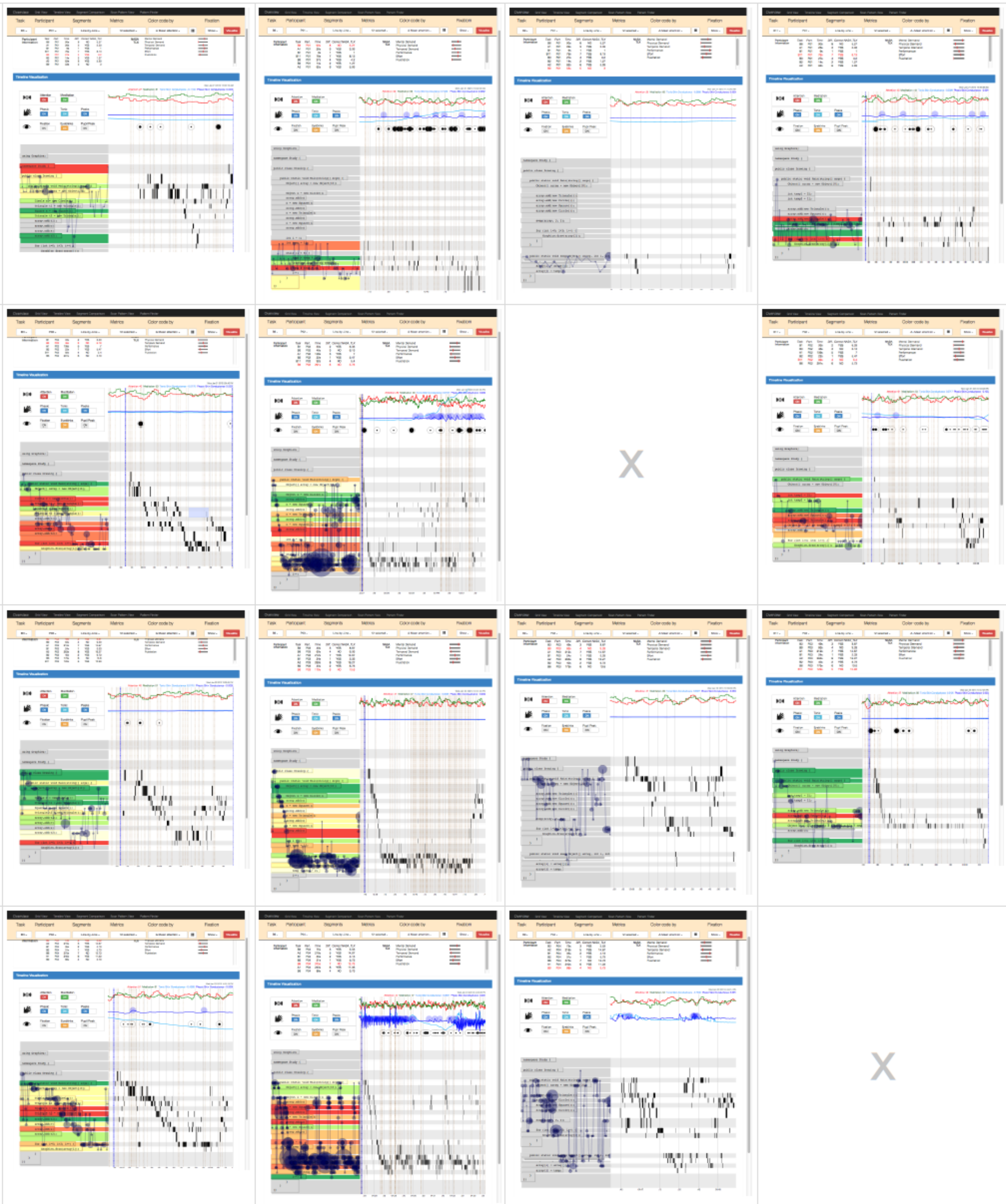
Appendix A

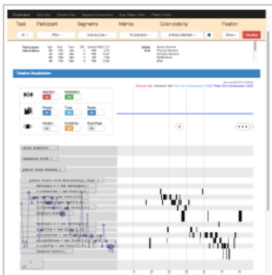
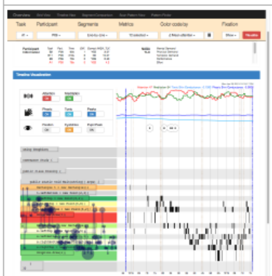
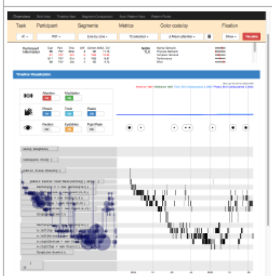
Visualization Screens

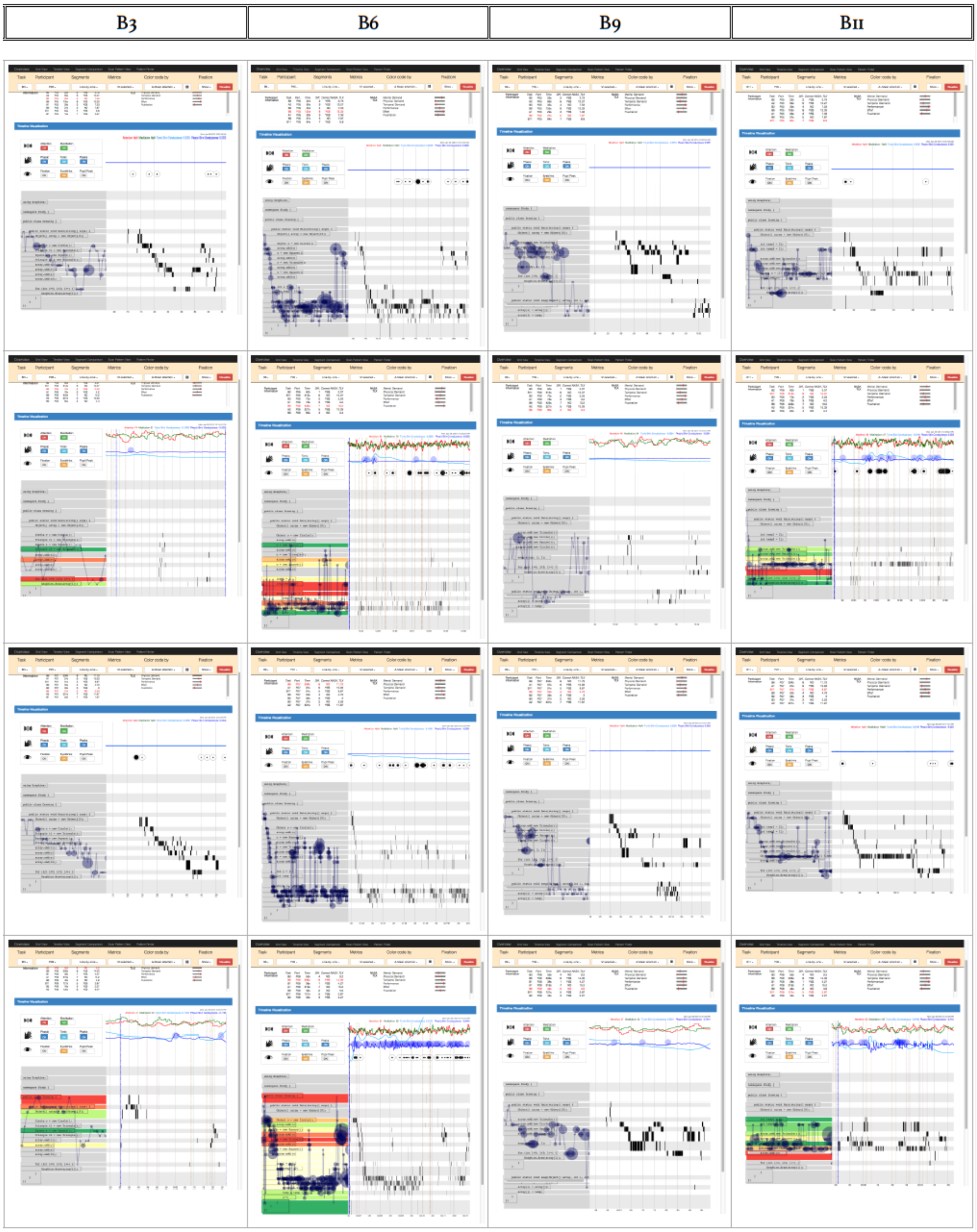
A.1 Timeline View Screens

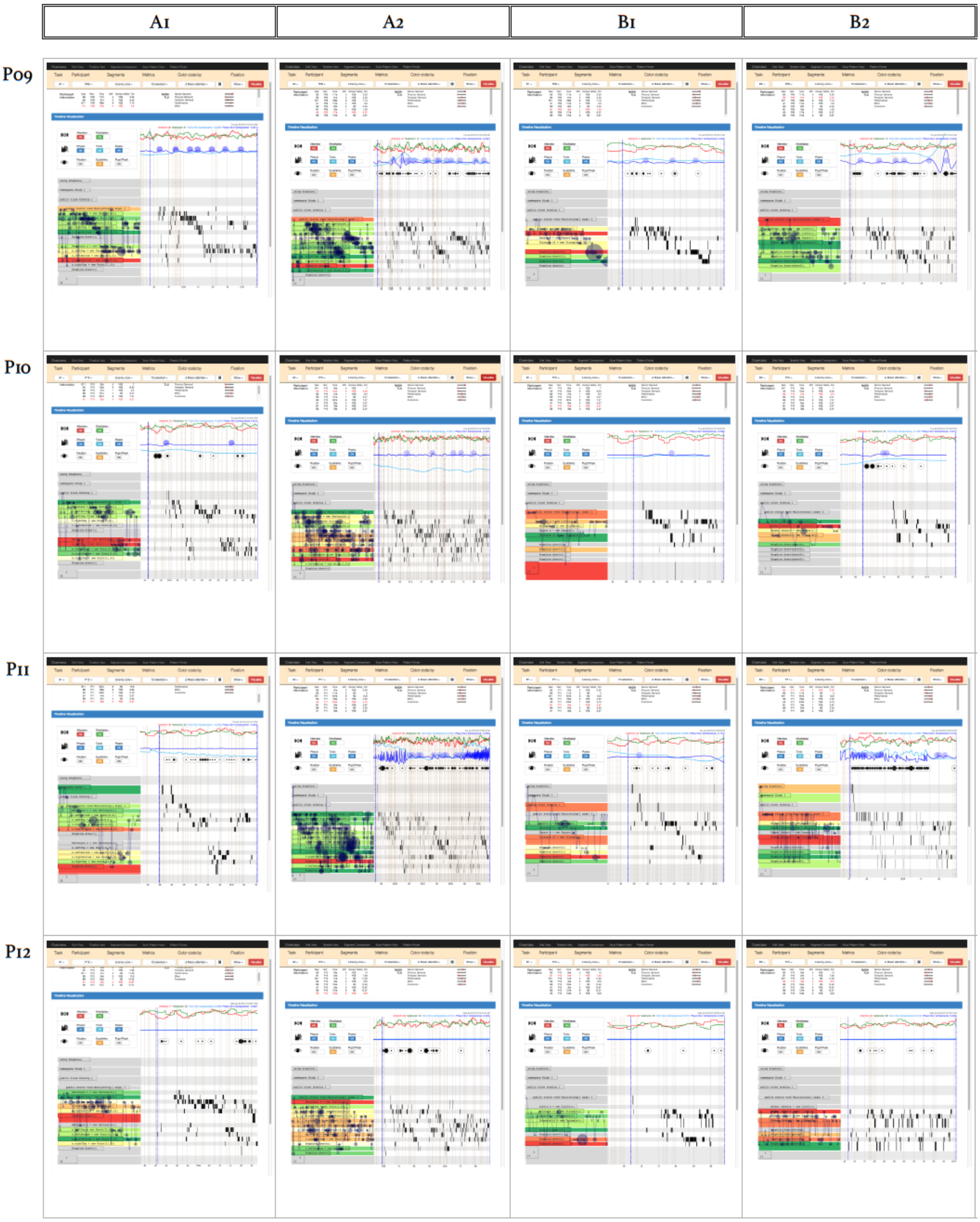
	A1	A2	B1	B2
P01				
P02		X		
P03				
P04				

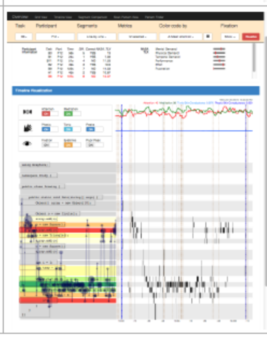
B3	B6	B9	B11



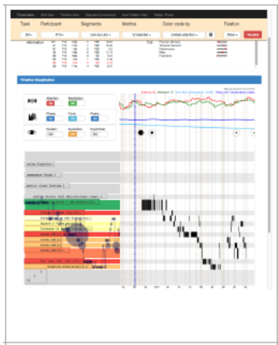
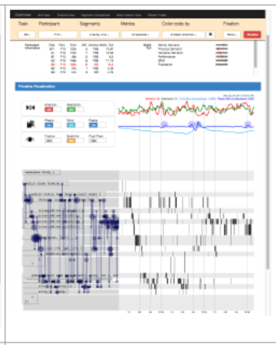
	A1	A2	B1	B2
P05				
P06			X	
P07				
P08				


B3
B6
B9
B11

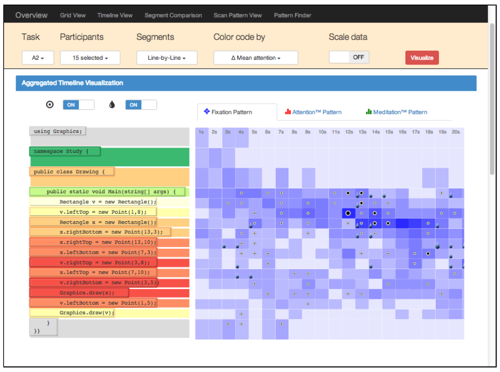
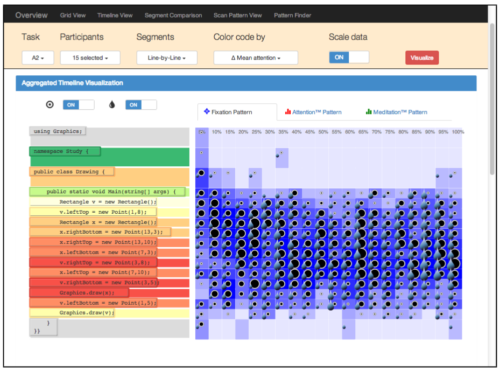

A1
A2
B1
B2
P09
P10
P11
P12

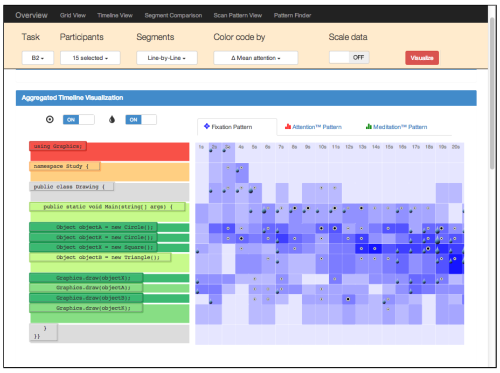
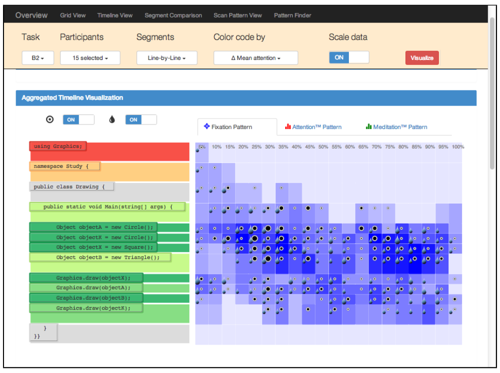
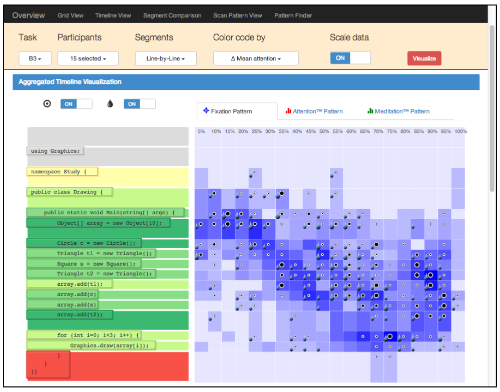
B3	B6	B9	B11
			
			
			
			

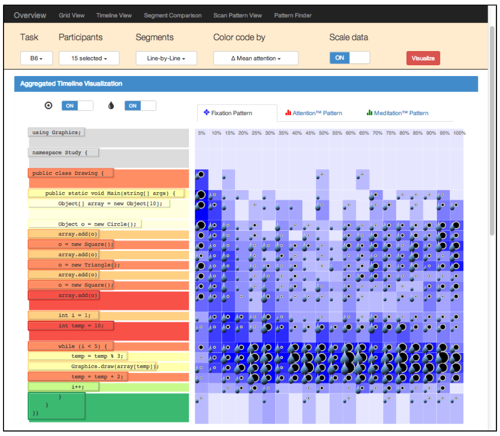
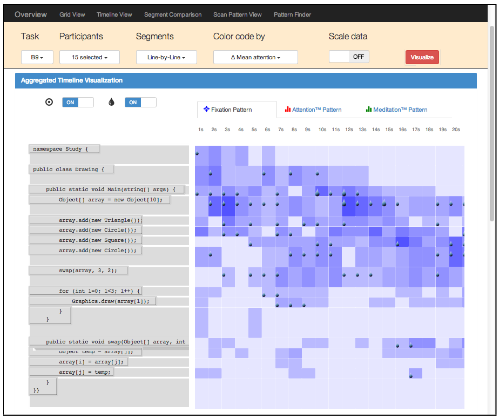
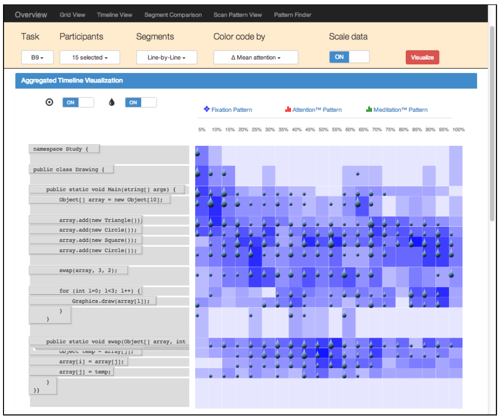
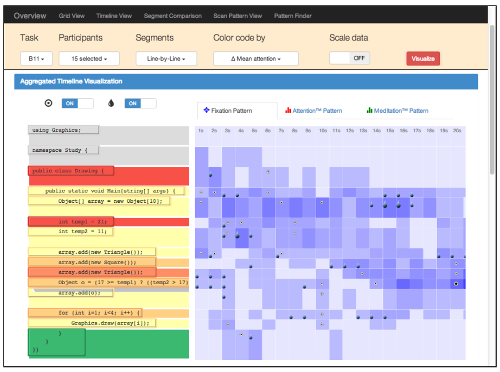
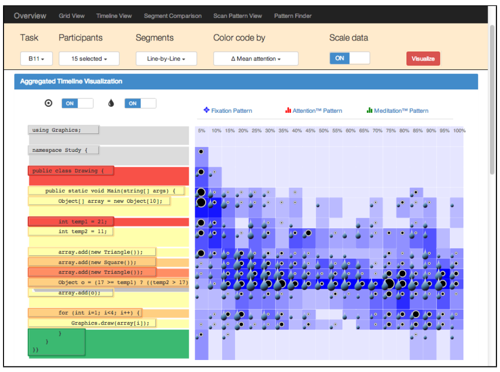
	A1	A2	B1	B2
P13				
P14				
P15				

B3	B6	B9	B11
			
			
			

A.2 Aggregated Timeline View Screens

	First 20s	Scaled View
A1		
A2		

	First 20s	Scaled View
B1		
B2		
B3		

	First 20s	Scaled View
B6		
B9		
B11		

Appendix B

Web Services

In this chapter, all the implemented web service operations that are used by the visualization application are listed. The string that has to be appended to the URI is specified in the tables for each web service operation. A complete URI of a sample web service call (Load fixation data for a specific participant and a specific AOI) is shown below:

Example: `http://{server-address}`**`/fixations/A1_AOILine8Hit/P04`**

B.1 Used by Grid View

Web services used by visualization module *Grid View*:

Operation	**Retrieve AOI measurments for Grid View**
HTTP Method	GET
URI	/features/{aoiName}/{participant(s)}
Request Example	http://localhost:8888/features/A1_AOILine8Hit/P06,P07
Request Body	None
Response Body	JSON representation of AOI measurement values
Description	Loads the AOI measurements (feature values) that are computed using Matlab. If in the Grid View multiple participants are selected, the all the metric values for all the participants are retrieved by one web service request. For that, the participant IDs are separated by commas. For each area of interest (AOI) that appears on the screen a web service request is required.

Table B.1: Web services that are used for the *Grid View*.

B.2 Used by (Aggregated) Timeline View

Web services used by the visualization module *Timline View* and *Aggregated Timline View*:

Operation	**Load fixation data**
HTTP Method	GET
URI	/fixations/{aoiName}/{participant}
Request Example	http://localhost:8888/fixations/A1_AOILine8Hit/P04
Request Body	None
Response Body	JSON representation of fixation data
Description	Loads a participants fixation data for a given area of interest. Returned are timestamp data for fixations that were recognized in a given area of interest. In addition to that, the gaze event duration is returned. Both information is used for the fixation bars and the fixation paths.

Operation	**Load eyeblink data**
HTTP Method	GET
URI	/eyeblinks/{task}/{participant}
Request Example	http://localhost:8888/eyeblinks/B1/P03
Request Body	None
Response Body	JSON representation of eyeblink data
Description	Loads timestamps that indicate when a eye blink has happened during the task. These timestamps are computed based on the EEG data using Matlab. In the visualization module, the eye blinks appear as dotted vertical lines on the timeline.

Operation	**Load Mindband data**
HTTP Method	GET
URI	/mindband/{task}/{participant}
Request Example	http://localhost:8888/mindband/A1/P01
Request Body	None
Response Body	JSON representation of attention and meditation values
Description	Loads the Attention and Meditation data that is displayed as a line chart integrated into the timeline visualization module.

Operation	**Load Pupil Peaks Data**
HTTP Method	GET
URI	/pupilPeaks/{task}/{participant}
Request Example	http://localhost:8888/pupilPeaks/B2/P10
Request Body	None
Response Body	JSON representation of pupil diameter peak data
Description	Loads timestamps of pupil peaks along with addition information about the peak (i.e. pupil size of dominant eye, size of peak, etc.)

Operation	**Load EDA data**
HTTP Method	GET
URI	/gsrTonicPhasic/{task}/{participant}
Request Example	http://localhost:8888/gsrTonicPhasic/A1/P01
Request Body	None
Response Body	JSON representation of phasic and tonic component of the EDA signal
Description	Loads the phasic- and tonic signal data that is displayed as a line chart integrated into the timeline visualization module.

Operation	**Load Phasic Peak Locations**
HTTP Method	GET
URI	/edaPeaks/{task}/{participant}
Request Example	http://localhost:8888/edaPeaks/B2/P10
Request Body	None
Response Body	JSON representation of the phasic peak locations
Description	Loads timestamps that indicate the location of the peaks on the timeline. In the timeline visualization, this data is used to highlight the peaks in the linechart.

Table B.2: Web services that are used for the *Timeline View*.

B.3 Used by Information Bar

Web services used by the information module:

Operation	**Retrieve data for information module**
HTTP Method	GET
URI	/info/{participant(s)}/{task(s)}
Request Example	http://localhost:8888/info/P01/all
Request Body	None
Response Body	JSON representation of task information facts
Description	Loads task information data such as NASA TLX values, task order, information whether a specific task was solved correctly or not. The keyword "all" can be used to retrieve i.e. all the statistical task information for a specific participant. To retrieve data about multiple participants, the participant IDs can be separated by commas.

Table B.3: Web services that are used for the *Information Bar*.

Appendix C

AOI Metrics

			Difficulty indicated by		
ID	Category	AOI Metric Name	High val.	Low val.	High abs. val.
1	EDA	Δ Mean tonic		x	
2	EDA	Δ Number of phasic peaks per s		x	
3	EDA	Δ Mean phasic peak slope		x	
4	EDA	Δ Mean phasic peak amplitude		x	
5	EDA	Min phasic peak amplitude	x		
6	EDA	Max phasic peak amplitude	x		
7	EDA	Δ Sum phasic peak ampl. per s		x	
10	Mindband	Min attention	x		
11	Mindband	Δ Median attention		x	
12	Mindband	Max attention	x		
13	Mindband	Δ Mean attention		x	
14	Mindband	Δ Stdev attention			
15	Mindband	Min meditation		x	
16	Mindband	Δ Median meditation	x		
17	Mindband	Max meditation		x	
18	Mindband	Δ Mean meditation	x		
19	Mindband	Δ Stdev meditation			
20	EEG	Δ Alpha/Beta	x		
21	EEG	Δ Alpha/Gamma	x		
22	EEG	Δ Alpha/Delta	x		
23	EEG	Δ Alpha/Theta	x		
24	EEG	Δ Beta/Alpha		x	
25	EEG	Δ Beta/Gamma			x
26	EEG	Δ Beta/Delta			x
27	EEG	Δ Beta/Theta			x
28	EEG	Δ Gamma/Alpha		x	
29	EEG	Δ Gamma/Beta			x
30	EEG	Δ Gamma/Delta			x
31	EEG	Δ Gamma/Theta			x
32	EEG	Δ Delta/Alpha		x	
33	EEG	Δ Delta/Beta			x
34	EEG	Δ Delta/Gamma			x
35	EEG	Δ Delta/Theta			x
36	EEG	ΔTheta/Alpha		x	
37	EEG	ΔTheta/Beta		x	
38	EEG	ΔTheta/Gamma		x	
39	EEG	ΔTheta/Delta		x	
40	EEG	Δ(Theta/Alpha)+Beta			
41	EEG	Δ(Beta/Alpha)+Theta		x	
50	Eyeblinks	Δ Eyeblinks per second		x	
60	Eyetracking	Fixations per second	x	x	
61	Eyetracking	Sum of fixation durations per s	x		
62	Eyetracking	Mean fixation duration	x		
63	Eyetracking	Median fixation duration	x		
64	Eyetracking	Stdev fixation duration			
70	Eyetracking	Min pupil size	x		
71	Eyetracking	Δ Median pupil size		x	
72	Eyetracking	Max pupil size	x		

Table C.1: List of AOI Metrics with corresponding implications regarding difficulty.

Appendix D

Experimental Study Tasks

D.1 Task A1

```
using Graphics;

namespace Study {

public class Drawing {

   public static void Main(string[] args) {
      Rectangle t = new Rectangle();
      t.leftBottom = new Point(2,2);
      t.leftTop = new Point(2,6);
      t.rightTop = new Point(6,6);
      t.rightBottom = new Point(6,2);
      Graphics.draw(t);

      Rectangle s = new Rectangle();
      s.leftTop = new Point(11,5);
      s.leftBottom = new Point(5,5);
      s.rightBottom = new Point(5,9);
      s.rightTop = new Point(11,9);
      Graphics.draw(s);

   }
}}

Will the two drawn rectangles overlap? yes / no
```

Listing D.1: Study Task A1.

D.2 Task A2

```
using Graphics;

namespace Study {

public class Drawing {

   public static void Main(string[] args) {
      Rectangle v = new Rectangle();
      v.leftTop = new Point(1,8);
      Rectangle x = new Rectangle();
      x.rightBottom = new Point(13,3);
      x.rightTop = new Point(13,10);
      x.leftBottom = new Point(7,3);
      v.rightTop = new Point(3,8);
      x.leftTop = new Point(7,10);
      v.rightBottom = new Point(3,5);
      Graphics.draw(x);
      v.leftBottom = new Point(1,5);
      Graphics.draw(v);
   }
}}

Will the two drawn rectangles overlap? yes / no
```

Listing D.2: Study Task A2.

D.3 Task B1

```
using Graphics;

namespace Study {

public class Drawing {

   public static void Main(string[] args) {

      Circle c = new Circle();
      Triangle t1 = new Triangle();
      Square s = new Square();
      Triangle t2 = new Triangle();

      Graphics.draw(t2);
      Graphics.draw(t1);
      Graphics.draw(c);
      Graphics.draw(s);

   }
}}


/*
*
* What are the last three shape objects drawn by Main()?
*
* (a) circle, triangle, square
* (b) triangle, square, circle
* (c) circle, triangle, triangle
* (d) triangle, circle, square
* (e) triangle, triangle, square
*
*/
```

Listing D.3: Study Task B1.

D.4 Task B2

```
using Graphics;

namespace Study {

public class Drawing {

  public static void Main(string[] args) {

     Object objectA = new Circle();
     Object objectK = new Circle();
     Object objectX = new Square();
     Object objectB = new Triangle();

     Graphics.draw(objectX);
     Graphics.draw(objectA);
     Graphics.draw(objectB);
     Graphics.draw(objectK);

  }
}}


/*
*
* What are the last three shape objects drawn by Main()?
*
* (a) circle, triangle, circle
* (b) circle, triangle, square
* (c) triangle, circle, circle
* (d) triangle, square, circle
* (e) triangle, circle, square
*
*/
```

Listing D.4: Study Task B2.

D.5 Task B3

```


using Graphics;

namespace Study {

public class Drawing {

   public static void Main(string[] args) {
      Object[] array = new Object[10];

      Circle c = new Circle();
      Triangle t1 = new Triangle();
      Square s = new Square();
      Triangle t2 = new Triangle();
      array.add(t1);
      array.add(c);
      array.add(s);
      array.add(t2);

      for (int i=0; i<3; i++) {
         Graphics.draw(array[i]);
      }
   }
}}


/*
 *
 * What are the last three shape objects drawn by Main()?
 *
 * (a) circle, triangle, square
 * (b) triangle, square, triangle
 * (c) triangle, circle, square
 * (d) circle, square, triangle
 * (e) none of the above
 *
 */
```

Listing D.5: Study Task B3.

D.6 Task B6

```
using Graphics;

namespace Study {

public class Drawing {

   public static void Main(string[] args) {
      Object[] array = new Object[10];

      Object o = new Circle();
      array.add(o);
      o = new Square();
      array.add(o);
      o = new Triangle();
      array.add(o);
      o = new Square();
      array.add(o);

      int i = 1;
      int temp = 10;

      while (i < 5) {
         temp = temp % 3;
         Graphics.draw(array[temp]);
         temp = temp + 2;
         i++;
      }
   }
}}

/*
*
* What are the last three shape objects drawn by Main()?
*
* (a) triangle, square, circle
* (b) square, square, triangle
* (c) square, circle, square
* (d) circle, triangle, square
* (e) circle, square, triangle
*
*/
```

Listing D.6: Study Task B6.

D.7 Task B9

```
using Graphics;

namespace Study {

public class Drawing {

   public static void Main(string[] args) {
      Object[] array = new Object[10];

      array.add(new Triangle());
      array.add(new Circle());
      array.add(new Square());
      array.add(new Circle());

      swap(array, 3, 2);

      for (int l=0; l<3; l++) {
         Graphics.draw(array[l]);
      }
   }

   public static void swap(Object[] array, int i, int j) {
      Object temp = array[j];
      array[i] = array[j];
      array[j] = temp;
   }
}}

/*
 *
 * What are the last three shape objects drawn by Main()?
 *
 * (a) triangle, circle, circle
 * (b) circle, square, triangle
 * (c) triangle, circle, square
 * (d) circle, circle, square
 * (e) square, circle, triangle
 *
 */
```

Listing D.7: Study Task B9.

D.8 Task B11

```
using Graphics;

namespace Study {

public class Drawing {

   public static void Main(string[] args) {
      Object[] array = new Object[10];

      int temp1 = 21;
      int temp2 = 11;

      array.add(new Triangle());
      array.add(new Square());
      array.add(new Triangle());
      Object o = (17 >= temp1) ? ((temp2 > 17) ? new Trian Object o = (17 >= temp1) ? ((
          temp2 > 17) ? new Triangle() : new Square()) : ((temp1 < temp2) ? new Circle()
           : new Square());
      array.add(o);

      for (int i=1; i<4; i++) {
         Graphics.draw(array[i]);
      }
   }
}}

/*
*
* What are the last three shape objects drawn by Main()?
*
* (a) triangle, square, triangle
* (b) circle, square, circle
* (c) square, triangle, triangle
* (d) square, triangle, square
* (e) square, triangle, circle
*
*/
```

Listing D.8: Study Task B11.

Appendix E

Used Libraries and Tools

E.1 JavaScript Libraries

Below, a selection of the most important JavaScript libraries that are used for the front-end implementation:

- **Visualization:**
  http://d3js.org/
  https://github.com/Caged/d3-tip
- **UI Elements**:
  https://github.com/davidstutz/bootstrap-multiselect
  http://bootstrapformhelpers.com/
  http://bootstrap-switch.org

E.2 NodeJS Modules

NodeJS modules that are used to manage the database access are listed below:

- **node-mysql**:
  https://npmjs.org/package/node-mysql
- **mongodb**:
  https://npmjs.org/package/mongodb

E.3 Matlab Tools and Algorithms

Used Toolboxes and predefined algorithms are listed below:

- **Peak finder algorithm**:
  http://mathworks.ch/matlabcentral/fileexchange/25500-peakfinder/content/peakfinder.m
- **EDA Toolbox**:
  https://github.com/mateusjoffily/EDA/wiki
- **Database Toolbox**:
  http://mathworks.ch/products/database/

Appendix F

Contents of the CD-Rom

The following files can be found on the CD-ROM:

- **Zusfsg.txt**
  German version of the abstract of this thesis
- **Abstract.txt**
  English version of the abstract of this thesis
- **Masterarbeit.pdf**
  Copy of this thesis
- **VisApp.zip**
  The visualization prototype described in this thesis
  (includes front-end implementation and back-end implementation)
- **DataProcessing.zip**
  Matlab M-Files and SQL-scripts
- **VisualizationOutputs.zip**
  Screenshot collection of visualization outputs

Bibliography

[AÇÇ09] Ali Özgür Argunşah, Ali Baran Çürüklü, and Müjdat Çetin. Detection of eye blinks from eeg using hidden markov models. 2009.

[Aff14] Inc. Affectiva. What are tonic and phasic changes? @ONLINE, June 2014.

[Ax53] Albert F Ax. The physiological differentiation between fear and anger in humans. *Psychosomatic Medicine*, 15(5):433–442, 1953.

[Bea82] Jackson Beatty. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological bulletin*, 91(2):276, 1982.

[BI08] Brian P Bailey and Shamsi T Iqbal. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(4):21, 2008.

[Bou12] Wolfram Boucsein. *Electrodermal activity*. Springer, 2012.

[BWJR14] Jason Braithwaite, Derrick Watson, Robert Jones, and Mickey Rowe. A guide for analysing electrodermal activity (eda) skin conductance responses (scrs) for psychological experiments @ONLINE, June 2014.

[CH06] Yongqin Cui and Jan M Hondzinski. Gaze tracking accuracy in humans: Two eyes are better than one. *Neuroscience letters*, 396(3):257–262, 2006.

[Chr81] Margaret J Christie. Electrodermal activity in the 1980s: a review. *Journal of the Royal Society of Medicine*, 74(8):616, 1981.

[CS90] Martha E Crosby and Jan Stelovsky. How do we read algorithms? a case study. *Computer*, 23(1):25–35, 1990.

[CSPM10] Katie Crowley, Aidan Sliney, Ian Pitt, and Dave Murphy. Evaluating a brain-computer interface to categorise human emotional response. In *ICALT*, pages 276–278, 2010.

[Cur14] Sophie Curtis. Microsoft tracks programmers' brain waves to reduce software bugs @ONLINE, July 2014.

[DG11] Ahmet Cengizhan Dirican and Mehmet Göktürk. Psychophysiological measures of human cognitive states applied in human computer interaction. *Procedia Computer Science*, 3:1361–1367, 2011.

[DJM90] Peter J De Jong and Harald Merckelbach. Eyeblink frequency, rehearsal activity, and sympathetic arousal. *International journal of neuroscience*, 51(1-2):89–94, 1990.

[Doe57] Donald G Doehring. The relation between manifest anxiety and rate of eyeblink in a stress situation. Technical report, DTIC Document, 1957.

[DSF07] Michael E Dawson, Anne M Schell, and Diane L Filion. 7 the electrodermal system. *Handbook of psychophysiology*, page 159, 2007.

[Dug13] Diana Dugas. Mindwave esense data @ONLINE, April 2013.

[FBM+14] Thomas Fritz, Andrew Begel, Sebastian C Müller, Serap Yigit-Elliott, and Manuela Züger. Using psycho-physiological measures to assess task difficulty in software development. In *ICSE*, pages 402–413, 2014.

[FM11] Bernd Figner and Ryan O Murphy. Using skin conductance in judgment and decision making research. *A handbook of process tracing methods for decision research*, pages 163–184, 2011.

[FWL59] A Ford, CT White, and M Lichtenstein. Analysis of eye movements during free search. *JOSA*, 49(3):287–289, 1959.

[GK99] Joseph H Goldberg and Xerxes P Kotval. Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, 24(6):631–645, 1999.

[HBPR09] M Haak, S Bos, S Panic, and LJM Rothkrantz. Detecting stress using eye blinks and brain activity from eeg signals. *Proceeding of the 1st driver car interaction and interface (DCII 2008)*, 2009.

[HS88] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

[ia13] iMedical apps. The q sensor device@ONLINE, April 2013.

[Ins14] Linton Instrumentation. Biopac gsr100c - gsr eda galvanic skin response amplifie @ONLINE, June 2014.

[JK03] Robert JK Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4, 2003.

[JN11] Marieke Jepma and Sander Nieuwenhuis. Pupil diameter predicts changes in the exploration–exploitation trade-off: evidence for the adaptive gain theory. *Journal of cognitive neuroscience*, 23(7):1587–1596, 2011.

[KBH11] Iftikhar Ahmed Khan, Willem-Paul Brinkman, and Robert M Hierons. Do moods affect programmers' debug performance? *Cognition, Technology & Work*, 13(4):245–258, 2011.

[KC01] Aarlenne Z Khan and J.Douglas Crawford. Ocular dominance reverses as a function of horizontal gaze angle. *Vision Research*, 41(14):1743 – 1748, 2001.

[KKH08] Jeff Klingner, Rakshit Kumar, and Pat Hanrahan. Measuring the task-evoked pupillary response with a remote eye tracker. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 69–72. ACM, 2008.

[Kli10] Jeff Klingner. Fixation-aligned pupillary response averaging. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 275–282. ACM, 2010.

[Min13] NeuroSky MindBand. The q sensor device@ONLINE, April 2013.

[MKK+12] Daniel McDuff, Amy Karlson, Ashish Kapoor, Asta Roseway, and Mary Czerwinski. Affectaura: an intelligent system for emotional memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 849–858. ACM, 2012.

[MUOM09] Yuko Matsuda, Hidetake Uwano, Masao Ohira, and Ken-ichi Matsumoto. An analysis of eye movements during browsing multiple search results pages. In *Human-Computer Interaction. New Trends*, pages 121–130. Springer, 2009.

[Nak11] Shashank Nakate. Galvanic skin response @ONLINE, 2011.

[Neu09] NeuroSky. Neurosky's esense™ meters and detection of mental state @ONLINE, June 2009.

[Neu14] NeuroSky. esense(tm) meters state @ONLINE, June 2014.

[Par11] Chris Parnin. Subvocalization-toward hearing the inner thoughts of developers. In *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on*, pages 197–200. IEEE, 2011.

[PD10] Chris Parnin and Robert DeLine. Evaluating cues for resuming interrupted programming tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 93–102. ACM, 2010.

[Ras73] David C Raskin. Attention and arousal. *Electrodermal activity in psychological research*, pages 125–155, 1973.

[RMDMM+09] Genaro Rebolledo-Mendez, Ian Dunwell, Erika A Martínez-Mirón, María Dolores Vargas-Cerdán, Sara De Freitas, Fotis Liarokapis, and Alma R García-Gaona. Assessing neurosky's usability to detect attention levels in an assessment exercise. In *Human-Computer Interaction. New Trends*, pages 149–158. Springer, 2009.

[SAdA08] Miguel A Sovierzoski, Fernanda IM Argoud, and Fernando M de Azevedo. Identifying eye blinks in eeg signal analysis. In *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on*, pages 406–409. IEEE, 2008.

[SFM12] Bonita Sharif, Michael Falcone, and Jonathan I Maletic. An eye-tracking study on the role of scan time in finding source code defects. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 381–384. ACM, 2012.

[SG00] Dario D Salvucci and Joseph H Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78. ACM, 2000.

[SM10] Bonita Sharif and Jonathan I Maletic. An eye tracking study on camelcase and under_score identifier styles. In *Program Comprehension (ICPC), 2010 IEEE 18th International Conference on*, pages 196–205. IEEE, 2010.

[Tec13] Tobii Technology. Tobii tx300 eye tracker @ONLINE, April 2013.

[Tep02] Michal Teplan. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.

[Tor07] Alexander Torstling. The mean gaze path: Information reduction and non-intrusive attention detection for eye tracking. 2007.

[UNMM06] Hidetake Uwano, Masahide Nakamura, Akito Monden, and Ken-ichi Matsumoto. Analyzing individual performance of source code review using reviewers' eye movement. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 133–140. ACM, 2006.

[Wag75] Michael J Wagner. Brainwaves and biofeedback a brief history—implications for music research. *Journal of Music Therapy*, 12(2):46–58, 1975.

[Wil02] Glenn F Wilson. An analysis of mental workload in pilots during flight using multiple psychophysiological measures. *The International Journal of Aviation Psychology*, 12(1):3–18, 2002.

[WSU14] Software Engineering Research Lab Wichita State University. itrace: An eye-tracking enabled software traceability environment @ONLINE, June 2014.

[Zü13] Manuela Züger. Exploring a developer's brain activity patterns. Master thesis, University of Zurich, Switzerland, August 2013.