# The Effect of Cognitive Load in Code Reading on Non-Programming Specific Environment

Hideaki Azuma*, Shinsuke Matsumoto*, Hidetake Uwano† and Shinji Kusumoto*
*Graduate School of Information Science and Technology, Osaka University, Japan
{h-azuma, shinsuke, kusumoto}@ist.osaka-u.ac.jp
†Department of Information Engineering, National Institute of Technology, Japan
uwano@info.nara-k.ac.jp

***Abstract*—Understanding program comprehension is one of the fundamental challenges of supporting software development. Although *code writing* is usually performed on programming specific environment, *code reading* is forced to be conducted in general environments such as physical paper. Our main hypothesis is that such a non-programming specific environment has some obstacles for program comprehension in terms of code presentation. The goal of this paper is to understand the effects on cognitive load caused by the obstacles. If our hypothesis will be proved and the goal will be achieved, we can provide the best practice of code presentation in non-programming specific environment.**

***Index Terms*—Program comprehension, syntax highlighting, typeface, EEG**

## I. Introduction

Understanding program comprehension is one of the fundamental challenges of supporting software development [1]. In the past decade, various studies [2] [3] [4] have investigated the developer's brain activity during program comprehension. Measuring brain activity may become a promising approach to reveal spontaneous biological reactions such as brain region activation [2], increase in cognitive load [3], and instinctive emotional response [4].

Although *code writing* is usually performed on programming specific environment (e.g., IDE), *code reading* is forced to be conducted in general environments, including physical paper, web browser, and PDF reader. For instance, researchers often read pseudo or example code written in an academic paper using a PDF reader. Most developers should have the experience to read source code through a web browser because the web is becoming a dominant information resource of programming knowledge.

Our main hypothesis is that such a non-programming specific environment has some obstacles for program comprehension in terms of code presentation. In contrast to general text documents, source code is displayed in a unique style that applied rich syntax highlighting and legible typeface. Syntax highlighting provides tiny but useful tips to distinguish source code elements visually [5]. A legible typeface is also an important presentation for the visibility of source code. A monospaced and non-serif typeface is typically used as a programming font. However, non-programming environments do not guarantee to show source code in proper styles. Automated and full syntax highlighting requires syntax analysis according to each programming language. Therefore, source code in academic papers, written using LaTeX and Microsoft Word, is often limited to highlight only reserved words. The worst case is that source code is provided with no-highlighting and proportional typeface.

The goal of this paper is to understand the effects on cognitive load caused by non-programming environments during code reading. We believe that the higher cognitive load occurs due to the lack of visibility which is supposed to be provided by syntax highlighting and legible typeface. We conduct a code reading experiment where subjects try to understand a given program in various reading environment to achieve the goal. The cognitive load is measured with an electroencephalogram (EEG) for each experiment trial.

## II. Code reading in non-programming environment

This paper defines a non-programming specific environment as a general text reading environment that is not specialized for programming tasks. Code reading, one of the programming tasks, is a distinct activity compared with general text reading. In code reading, a programmer firstly tries to grasp an overall code structure, understand variable relations, and follow the control flows. Proper source code presentation may help such comprehension processes.

This paper aims to investigate the effect of the code presentations for code reading. We examine the following three research questions.

- RQ1: Can EEG detect cognitive load in code reading?
- RQ2: Does syntax highlighting affect cognitive load?
- RQ3: Does typeface affect cognitive load?

Figure 1 shows the relation of the three research questions and compared code reading environments. The X-axis means three typefaces, and the Y-axis shows four types of highlighting. The compared reading environment is labeled as $\mathtt{E}^{\mathtt{highlight}}_{\mathtt{typeface}}$ . For example, $\mathtt{E}^{+}$ is the best reading environment where source code is displayed with fully highlighted and monospace typeface. We regard $\mathtt{E}$ as a baseline environment where only reserved words are highlighted with the monospaced typeface. $\mathtt{E}$ is often used in academic papers because of the simplicity of its application. The reserved word highlighting requires not syntax analysis but just a dictionary of keywords.

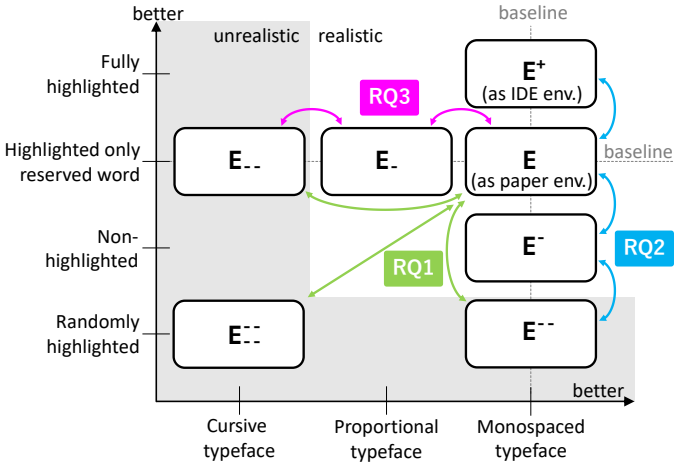


Fig. 1: Three RQs and compared seven environments.

We prepare three unrealistic environments where the code is shown with random highlighting ($\mathtt{E}^{--}$), cursive typeface ($\mathtt{E}_{--}$), and both of them ($\mathtt{E}^{--}_{--}$) to answer the RQ1. In the RQ1, EEG of the baseline environment ($\mathtt{E}$) is compared with each unrealistic environments ($\mathtt{E}^{--}$, $\mathtt{E}_{--}$, $\mathtt{E}^{--}_{--}$). RQ2 and RQ3 are confirmed by comparing highlightation ($\mathtt{E}^{+}$, $\mathtt{E}$, $\mathtt{E}^{-}$and $\mathtt{E}^{--}$) and typeface ($\mathtt{E}$, $\mathtt{E}_{-}$ and $\mathtt{E}_{--}$), respectively.

## III. Experiments

### A. Summary

The purpose of this experiment is to show the effect of syntax highlighting and typeface on EEG during code reading. The subjects are measured EEG until the subjects answer the output of a given program.

In the experiment, we use $\beta / \alpha$ which is the ratio of beta wave to alpha wave as an indicator to measure cognitive load caused by the code reading. The alpha wave and the beta wave are respectively normalized. This value has been indicated that it is valid for observing the state of thinking [6]. The higher this value indicates that the higher cognitive load occurs.

### B. Tasks

We set small Java programs which have a certain output as reading tasks. Each program has a comment which can be a clue for understanding. We prepare 30 programs including two practices. Besides, practice programs are easier than other programs. 28 programs are composed of four classes for every seven programs. The four classes are controlling list, controlling string, mathematics and conditional branch.

In the experiment, the order of code reading is counterbalanced to avoid the effect on the results, comprehending speed and habituation. We change the order of the programs and the order of the environment of them.

### C. EEG Measurement

A subject is attached EEG device to measure EEG during code reading. Electrodes are located according to the monopolar derivation method. A Ground electrode is located at the right ear (A2). A Standard electrode is located at the left ear (A1). Four measurement electrodes are located at the back of the head (Pz), the glabella (Fpz) and the forehead (F3, F4). EEG is easy to affect by myoelectric potential. Thus, We instruct subjects to suppress the movement.

### D. Experimental Procedure

(1) We explain the sequence of the experiment.
(2) We attach the EEG device to a subject. We measure the EEG of the subject with eyes closed before beginning the experiment to confirm whether we can measure the subject's EEG or not.
(3) The subject reads two practice programs to experience reading and answering the output. Subjects read the first practice program on $\mathtt{E}^{+}$ and the second program on $\mathtt{E}^{--}_{--}$ to experience the best and worst environments.
(4) The subject read the program on display. Then the subject thinks the output of it. We measure the subject's EEG during code reading.
(5) The subject tells us that he understood the output of the program. Then the subject answers the summary of program behavior to show their understanding.
(6) We repeat step (4) and (5) 28 times. The subject takes a one minute break for every seven reading to avoid the effect of tiredness to avoid the effect of tiredness.

## IV. Conclusion

In this paper, we show the experimental method for understanding the effect of cognitive load caused by syntax highlighting and typeface by using EEG.

One important future work is to conduct this experiment and confirm our hypothesis. In addition, combining other biometrics, such as eye movement, will provide further indicators of program comprehension.

## Acknowledgment

This work was supported by JSPS/MEXT KAKENHI Grant Number 18H03222.

## References

[1] N. Peitek, J. Siegmund, C. Parnin, S. Apel, and A. Brechmann, "Toward conjoint analysis of simultaneous eye-tracking and fmri data for program-comprehension studies," in *Proc. Workshop on Eye Movements in Programming*, 2018.
[2] Y. Huang, X. Liu, R. Krueger, T. Santander, X. Hu, K. Leach, and W. Weimer, "Distilling neural representations of data structure manipulation using fmri and fnirs," in *Proc. International Conference on Software Engineering*, 2019, pp. 396–407.
[3] S. Lee, A. Matteson, D. Hooshyar, S. Kim, J. Jung, G. Nam, and H. Lim, "Comparing programming language comprehension between novice and expert programmers using eeg analysis," in *Proc. International Conference on Bioinformatics and Bioengineering*, Oct 2016, pp. 350–355.
[4] Y. Lin, C. Wang, T. Jung, T. Wu, S. Jeng, J. Duann, and J. Chen, "Eeg-based emotion recognition in music listening," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 7, pp. 1798–1806, 2010.
[5] C. Hannebauer, M. Hesenius, and V. Gruhn, "[journal first] does syntax highlighting help programming novices?" in *Proc. International Conference on Software Engineering*, May 2018, pp. 704–704.
[6] F. Hirai, K. Yoshida, and I. Miyaji, "Comparison analysis of the thought and the memory at the learning time by the simple electroencephalograph," *Proc. Multimedia, Distributed, Cooperative, and Mobile Symposium*, pp. 1441–1446, 2013.