

構文ハイライトとフォントの違いによる開発者の脳波の比較

東 英明[†] 梶本 真佑[†] 上野 秀剛^{††} 楠本 真二[†]

[†] 大阪大学大学院情報科学研究科

^{††} 奈良工業高等専門学校

E-mail: [†]{h-azuma,shinsuke,kusumoto}@ist.osaka-u.ac.jp, ^{††}uwano@info.nara-k.ac.jp

あらまし 開発者がどのようにプログラムを理解するのかという疑問の解明はソフトウェア工学の重要な課題の一つである。プログラム記述は通常 IDE などのプログラミング特化環境で行われるが、プログラム読解は紙などのプログラミングに特化していない環境で行われることもある。そのような非プログラミング特化環境のプログラム読解はプログラミング特化環境でのプログラム読解より困難である事があるため、非プログラミング特化環境でのプログラム提示にはプログラム理解の障害となる要素が存在すると考えた。本稿の目的は、そのような障害要素に起因する認知負荷の影響の理解にある。この目的を達成するために、我々はよりプログラムに適していない環境でのプログラム読解ほど、大きな認知負荷が発生するという仮説のもと、実験を行った。しかし、実験結果は我々の仮説とは異なる結果を示していた。そのため本稿では、実験結果が仮説に反する結果となった理由を考察する。

キーワード プログラム理解, シンタックスハイライト, フォント, 脳波

1. はじめに

開発者がどのようにプログラムを理解するのかという疑問の解明はソフトウェア工学の重要な課題の一つである [1]。近年では、精度や速度といった比較的観測が容易な尺度を用いた調査に限らず、生体計測を用いた開発者の内発的な反応を調べる研究も増えている [2] [3] [4]。例えば、脳波と脳血流によるデータ構造把握時の脳活動領域の調査 [2]、脳波によるプログラム理解時の認知負荷の増減の調査 [3]、脳波と眼球運動の同時測定によるプログラム理解中の生体反応の分析 [4] などの研究がある。このように、脳活動の計測に基づく調査は、被験者自身も気付かない生体的な反応や、認知負荷の大きさを客観的に示すことができる。

また、プログラム記述は通常 IDE などのプログラミング特化環境で行われるが、プログラム読解は、紙面などのプログラミングに特化していない環境で行われることもある。例えば、研究者は論文内の疑似コードや例題プログラムを PDF リーダーなどを使用して読むことも多い。近年では、プログラム開発の情報源として Web が広く使用されるようになった。そのため、ブラウザ上でのサンプルコードの読解は多くの開発者に経験があると考えられる。

本稿の調査の主たる疑問は「非プログラミング特化環境でのプログラム読解にはプログラム理解の障壁となる要素があるのではないか」という点にある。プログラミング特化環境では、ソースコードは予約語や変数名など、トークンの意味によって色分けされたシンタックスハイライトを施されることが多い。そのようなハイライトの適用は、構文的な誤りの発見が容易になるだけでなく、制御の構造の直感的な視認を可能にする。さらに、プロ

グラミング特化環境では、等幅フォントのようなプログラムに適したフォントが使用される事が多い。等幅フォントの使用は、“o”と“0”、“1”と“l”などの形の似た文字を判別しやすくすることで、読み間違いを減らすことができる。一方、ソースコードに非等幅フォントを使用すると、インデントを活用した際に行頭のずれが起きてしまう。インデントは制御の流れを視覚的に理解しやすくするため、インデントがずれることによりソースコードの視認性が下がってしまう。よって、適切なフォントの使用により、ソースコードの判読性や視認性を維持できる。

しかし、非プログラミング特化環境ではハイライトやフォントが適切な状態でソースコードが表示されているとは限らない。ソースコードに適切なハイライトが施されない理由は、自動的なシンタックスハイライトが、適用対象となるプログラミング言語それぞれに応じた構文解析が必要なためである。また、ソースコードに適切なフォントが使用されない理由は、多くの文書作成ソフトのデフォルトのフォントが非等幅フォントであるためである。そのため、 $\text{\LaTeX}$ や MS Word を用いて記述された論文内のソースコードはハイライトが予約語に限定されていることが多い。さらに、ハイライトが全く施されておらず、非等幅フォントを使用してソースコードが記述されている場合さえある。

本稿の目的は、非プログラミング特化環境特有の問題に起因するプログラム読解時の認知負荷の影響の理解にある。まず、シンタックスハイライトとフォントがプログラムに不適な状態になると、プログラム読解が困難になるのではないかと考え、非プログラミング特化環境ではプログラム読解時の認知負荷が増大するという仮説を立てた。そこで、その仮説を検証するため、11 人の被験者に様々な環境でプログラムを読解させることによ

り、その間の認知負荷を脳波で調査する実験を行った。しかし、実験では我々の仮説とは異なる結果が確認された。従って本稿では、実験結果が仮説に反する結果となった理由を考察する。

2. 関連研究

2.1 プログラム理解と脳波

脳波は感情や精神的な状態を測る手段として広く使用されている [5]。脳波の中でも、8~13Hz は α 波と呼ばれ、リラックスなどの落ち着いた心理状態に関係があるとされている。また、13~30Hz は β 波と呼ばれ、集中や思考などの精神活動に関係があるとされている。さらに、脳波はプログラム理解の研究にも多く用いられている [3] [6]。Lee らはプログラミングの初心者と熟練者のプログラム理解の能力差を脳波によって調査し、結果を示した [3]。また、Yeh らは難読化されたプログラムの読解中の脳波を計測し、認知負荷を調査する実験を行っている [6]。本稿では、彼らの研究と同様に、プログラム読解中の認知負荷を脳波によって計測する。

2.2 プログラム理解とプログラムの環境

シンタックスハイライトやフォント、インデントなどのプログラムの環境がプログラム理解にとって重要な要素の一つであることは広く知られている。IDE のようなプログラミング特化環境は通常、プログラムに最も適したハイライトやフォントで使用者にプログラムを提示する。しかし、非プログラミング特化環境ではその限りでなく、ハイライトやフォントがプログラムにとって適切である保証はない。また、シンタックスハイライトのプログラム理解への影響を調査した研究もいくつか存在している [7] [8] [9]。その結果、Sarkar らは、シンタックスハイライトが施されていることで、理解に要する時間が短くなると結論づけている [7]。しかし、Hannebauer らの研究では、プログラミング初心者にはシンタックスハイライトが施されていても、読解時間が減ることがなかったと結論付けられている [8]。また、Beelders らの研究でも、シンタックスハイライトがプログラミングに有効であることを示すことはできなかったと結論づけられており [9]、シンタックスハイライトの効果に対する結論は研究によって様々であった。本稿でも、シンタックスハイライトの影響を調査するが、彼らの研究との相違点は、脳波を使用してハイライトやフォントが認知負荷に与える影響を調査することにある。

3. 非プログラミング特化環境でのプログラム読解

本稿では、ウェブサイトや論文などのように、プログラムが記載されることが想定されていない環境を非プログラミング特化環境として定義する。プログラミングの際に行う作業の一つである、プログラム読解は一般のテキストの読解とは全く異なる性質を持っている。プログラム読解では、プログラマはまず構造の全体像を掴もうとすることが多い。そして、それに続いて、変数の関係性を理解し、処理の流れを理解するという流れで理解が進んでいく。プログラミング特化環境のような適切なソースコードの提示は前述のプログラム理解の流れを円滑にする。

本稿の目的は、プログラム読解時の提示されたプログラムの

環境の差による認知負荷への影響を調査することにある。そこで、我々は以下の 3 つの Research Question を立てた。

RQ1: 脳波によるプログラム読解中の認知負荷の計測が可能か。

RQ2: シンタックスハイライトは認知負荷に影響を及ぼすのか。

RQ3: フォントは認知負荷に影響を及ぼすのか。

上記 3 つの Research Question と本稿の実験で比較された 7 つのプログラムの環境との対応を図 1 に示す。横軸は 3 種類のフォントを表しており、縦軸は 4 種類のシンタックスハイライトを表している。図に示されたそれぞれの環境の例のプログラムを図 2 に示す。図 2 の縦軸と横軸は図 1 と同等のものである。また、それぞれの環境は `Ehighlightingtypface` のように表されている。例えば、`E+` は IDE 並みの意味的な情報が豊富なハイライトが施されており、フォントには等幅フォントが使用されている、最もプログラムを読むのに適した環境である。また、本稿では、予約語のみハイライトされており、等幅フォントが使用されている、`E` をベースラインとしている。予約語のみのハイライトを適用する際には構文解析を必要とせず、予約語のリストさえあれば簡単に施すことができる。そのため、`E` は適用の容易さから、学術論文上のプログラム例などによく見られる例である。

RQ1 において、プログラム読解中の認知負荷を脳波で測ることができるかを確認するため、非現実的なレベルでプログラムの読解が困難になるような環境を設定する必要があった。そのため、本稿では、ハイライトが非現実的である、トークン毎にランダムなハイライトが施されている `E-`、フォントが非現実的である、筆記体のフォントでプログラムが記述されている `E-`、ハイライトとフォントが両方非現実的である `E--` を設定している。RQ1 では、ベースラインである `E` と `E-`、`E-`、`E--` という 3 つの非現実的な環境との比較によって確認を行う。RQ2 ではハイライトの認知負荷への影響を調べるため、ベースラインである `E` とハイライトのみが異なっている、`E+`、`E-`、`E--` との比較によって行う。RQ3 ではフォントの認知負荷への影響を調べるため、ベースラインである `E` とフォントのみが異なっている、`E-`、`E--` との比較によって行う。しかし、我々の分析では、RQ1 の結果でさえ有意差がない、ネガティブな結果であった。よって、RQ1 で認知負荷が脳波によって計測できる前提の上に成り立っている、RQ2 と RQ3 にのみ関係し、中間的な環境である `E-` と `E-` はページ制限を考慮し、この章以降では除外して考察を行う。

4. 実験

4.1 実験概要

この研究の目的はシンタックスハイライトとフォントがプログラム読解によって起こる認知負荷に及ぼす影響を示すことである。実験は被験者が与えられたプログラムの出力を答えるまでの間の脳波を測定することにより行われる。本実験の被験者は 10 人の大阪大学の学生と、1 人の教員であり、全員 Java の使用歴が 1 年以上であった。

この研究では、プログラム読解時に起こる認知負荷を測定する分析指標として α 波と β 波の割合である、 β/α を使用する。この α 波と β 波はそれぞれ被験者の全プログラム読解の α 波と β 波の平均値で正規化されている。また、この β/α という値

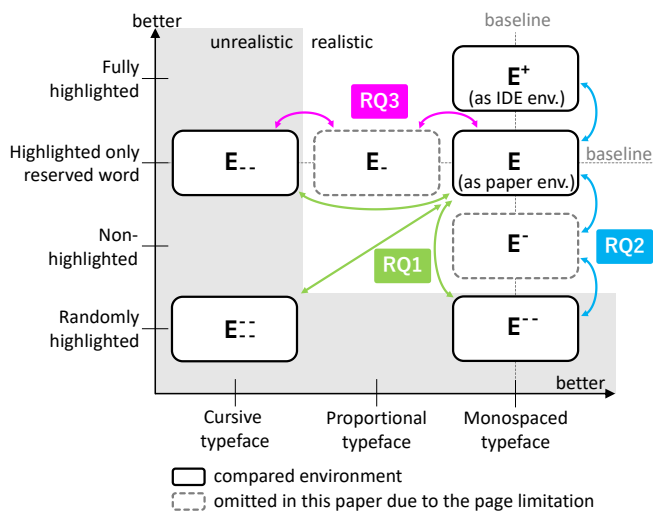


図 1 RQ と比較対象である 7 つの環境との対応づけ

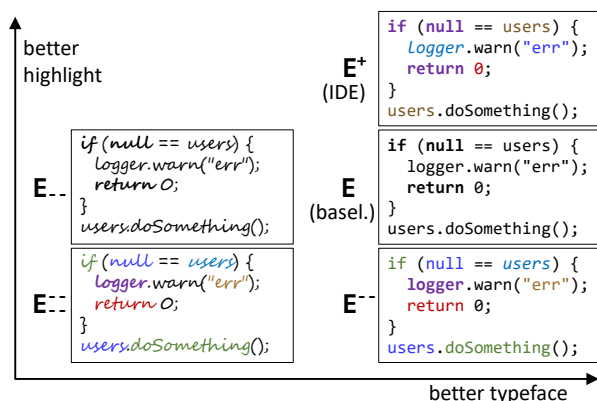


図 2 それぞれの環境のプログラム例^(注1)



図 3 実験時の様子

は、人の思考状態を観察するために有効であることが示されており [10], この値が高ければ高いほど、より大きな認知負荷がかかっていることを示す。

4.2 実験環境

図 3 に実験時の被験者の様子を示す。実験は被験者 1 名と実験者 1 名のみがいる静かな部屋で実施する。脳波の計測装置はナノテックイメージ社製 NeXus-10 MARKII を用いる。本計測装置の計測周期は 256Hz である。計測された脳波は Bluetooth 経由で脳波計測用 PC に転送され、CSV ファイル形式で出力さ

れる。

実験者は 1 台の PC で装置の制御を行い、もう 1 台の PC でデータの記録、及び問題の提示をする。被験者には PC に接続した別のディスプレイでプログラムを提示する。被験者は提示されたタスクの出力を頭の中で考え、分かった時点でその旨を実験者に伝える。脳波以外の体動によるノイズ（アーチファクト）を抑制するため、被験者は実験中にキーボードやマウスなどの操作は一切行わない。また、同様の理由から、被験者は肘掛けを備えた椅子に座り、椅子の高さやディスプレイの高さ、ディスプレイとの距離を事前に調節する。

4.3 実験タスク

実験タスクとして、1 つの出力をもつ Java プログラムを読み、その出力を答えるというタスクを被験者に課す。それぞれのプログラムは処理の内容を理解する手がかりとなるコメントを含む。そのようなプログラムを練習用に 2 題、本番用に 28 題の合計 30 題を用意する。

出題するプログラムはできるだけ難易度が揃うよう、プログラミングコンテストである AtCoder^(注2) の B 問題から C 問題の内容を参考に改題し、作成した。参考にした B 問題や C 問題の B, C はそれぞれ問題の難易度を表しており、これらの問題はいくつもの if 文や for 文を用いることによって解くことができる程度の難易度である。今回プログラム問題の作成にこれらの問題を参考にした理由は、これらの問題から作成されるプログラムが簡単すぎず、プログラミングコンテストの同じランクの問題であればある程度難易度が均一になっているのではないかと考えたためである。また、練習用プログラムは本番用プログラムよりも簡単な難易度に設定し、実験の最初に全被験者に練習用プログラムをそれぞれ E+, E- の環境の順で提示した。本番用プログラムは以下の 4 つの系統の問題を各 7 問用意する。

- リスト操作系
- 文字列操作系
- 数学系
- 条件分岐系

なお、タスクは全て GitHub のリポジトリ^(注3) 上で公開されている。

今回の実験では、制限時間の存在により、被験者が焦り、十分にプログラムを理解できない可能性があることを考慮し、制限時間は設けずに各タスクを実施した。しかし、プログラムを質問する上では最大でも 5 分以内で出力を導けることを想定した。また、実験中は前の問題の誤答による被験者の心理的な不安を除くため、回答の正誤は伝えない。さらに、この実験では、プログラムの提示順序が脳波や理解の速度、被験者の慣れに影響することから、その順序効果を緩和するためにカウンターバランスを行う。具体的には被験者に提示するプログラムの順序、プログラムの環境の順序を被験者毎に変化させ、どの被験者も全く同じ順序で同じ環境のプログラムが提示されないようにした。

(注1) : E- と E- は除外してある。

(注2) : <https://atcoder.jp/>

(注3) : https://github.com/kusumotolab/h-azuma_experiment-tasks

4.4 脳波計測

練習タスクの開始前に被験者に脳波計測装置を装着し、読解中の脳波を計測する。また、プログラム読解の回数の多さから被験者の疲労を考慮し、本番タスク 7 回に 1 回のペースで 1 分間の休憩を取りながら行った。

脳波の計測に用いる電極は、グラウンド電極を右耳 (A2) とし、基準電極導出法 (片側耳朶法) を用いて、基準電極を左耳 (A1)、計測用電極を肩間に 1 箇所 (Fpz)、前頭部に 2 箇所 (F3, F4)、後頭部に 1 箇所 (Pz) に配置した。計測用電極を肩間 (Fpz)、前頭部 (F3, F4) に配置した理由は、感情的な脳波の変化が前頭部に出やすいとされているためである。また、残りの計測電極を後頭部 (Pz) に配置した理由は、後頭部は筋電位など脳波以外の生体現象のアーチファクトが入りにくいためである。電極の装着不良によるアーチファクトを防ぐため、被験者には頭部用ネット包帯を着用してもらう。また、脳波は筋電位によるノイズに影響を受けやすいため、実験中はできるだけ体を動かさないように被験者に指示する。

4.5 実験手順

実験の手順を以下に示す。

(1) 実験説明・準備

実験の流れについての説明と、脳波計測時の注意を行う。

(2) 装置の設定

被験者に電極を配置し、脳波計測装置の設定を行う。脳波が正常に測れているかを確認するために閉眼安静状態の脳波を計測する。

(3) 練習タスクの実施

被験者に問題の雰囲気と答え方を経験させるため、練習タスクを 2 回行う。それぞれのプログラムは全被験者共通で、1 問目は E⁺、2 問目は E[−] の環境で提示する。

(4) 本番タスクの実施

ディスプレイにプログラムを表示させ、その Java プログラムから得られる出力を考えてもらう。被験者は出力を理解した時点で声を出して実験者に知らせる。また、プログラムを読んでいる間の脳波を計測する。

(5) 口頭での回答

実験者にプログラムの出力が分かった旨を伝えたあと、被験者は出力の内容を答える。その後、続けてそのプログラムの処理の内容を答える。

(6) 全タスクの実施

手順 4, 5 を 28 回繰り返し、7 問に一回の間隔で 1 分間の休憩を挟む。

4.6 分析方法

分析はタスクの開始直後 5 秒間の FFT 結果と被験者が声を上げるタスク終了時直前の 5 秒間の FFT 結果の差分を取るによって行う。これにより、長時間の実験のために起こった電極のずれによるノイズの影響を小さくできる。肩間 (Fpz) 及び前頭部 (F3, F4) から得られた脳波には眼球運動や瞬きが原因と考えられるフィルタで除去できない程の大きなノイズがのっていたため、分析にはそれぞれの被験者の後頭部 (Pz) の電極から得られた脳波データを使用する。分析の流れを以下に示す。

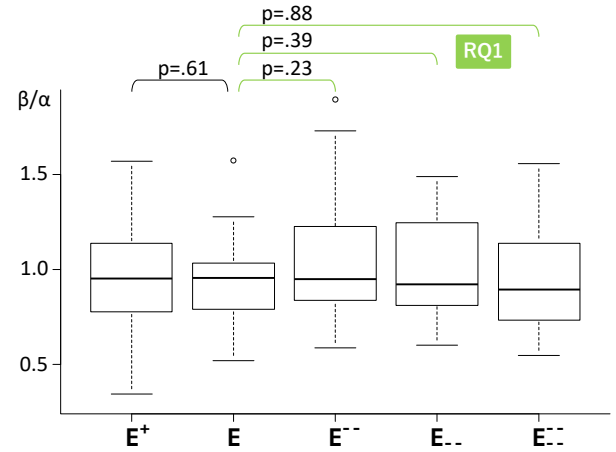


図 4 各環境における全被験者の β/α の比較

(1) ノイズ除去

計測した脳波に閾値が 45Hz のローパスフィルタの適用により、体動、眼球運動などのアーチファクトの除去を行う。

(2) 開始直後 5 秒間、終了直前 5 秒間の脳波の抽出

被験者 s がタスク n を実施した際のノイズ除去済みの脳波を $BW(s, n)$ とする。ここで、各タスクにおいて時間 t が $0 \leq t \leq 5$ の $BW(s, n)$ を開始直後 5 秒間の脳波として抽出する。抽出した開始直後 5 秒間の脳波を $BW_s(s, n)$ とする。また、各被験者がタスク n のプログラムの理解にかかった時間を T 秒とし、 $T - 5 \leq t \leq T$ の $BW(s, n)$ を終了直前 5 秒間の脳波として抽出する。抽出した終了直前 5 秒間の脳波を $BW_e(s, n)$ とする。

(3) フーリエ変換・差分の取得

計測周期 256/sec より $N=256$ で $BW_s(s, n)$ と $BW_e(s, n)$ に対し FFT をかけ、 $BW_s(s, n)$ と $BW_e(s, n)$ の FFT 結果の差分を取り、その差分のパワースペクトルを得る。パワースペクトルを求める周波数範囲は α 波と β 波として定義されている周波数帯域である 8Hz~30Hz とする。

(4) 成分抽出

得られたパワースペクトルから、 α 波の帯域である 8Hz~14Hz、 β 波の帯域である 14Hz~30Hz、それぞれの帯域成分を抽出する。

$$\alpha(BW_s(s, n) - BW_e(s, n))$$

$$\beta(BW_s(s, n) - BW_e(s, n))$$

(5) 正規化

脳波は個人差が大きいため、抽出された成分データを被験者の平均値で正規化する。ここで、 N は本番タスクの回数とする。

$$Norm\alpha(s, n) = \frac{\alpha(BW_s(s, n) - BW_e(s, n))}{\frac{1}{N} \sum_{n=1}^N \alpha(BW_s(s, n) - BW_e(s, n))}$$

$$Norm\beta(s, n) = \frac{\beta(BW_s(s, n) - BW_e(s, n))}{\frac{1}{N} \sum_{n=1}^N \beta(BW_s(s, n) - BW_e(s, n))}$$

5. 実験結果

5.1 プログラム読解時の脳波

まず、結果の全体像として環境毎の β/α を比較する。環境毎のタスク中の β/α を図 4 に示す。横軸は各環境を表し、縦軸は

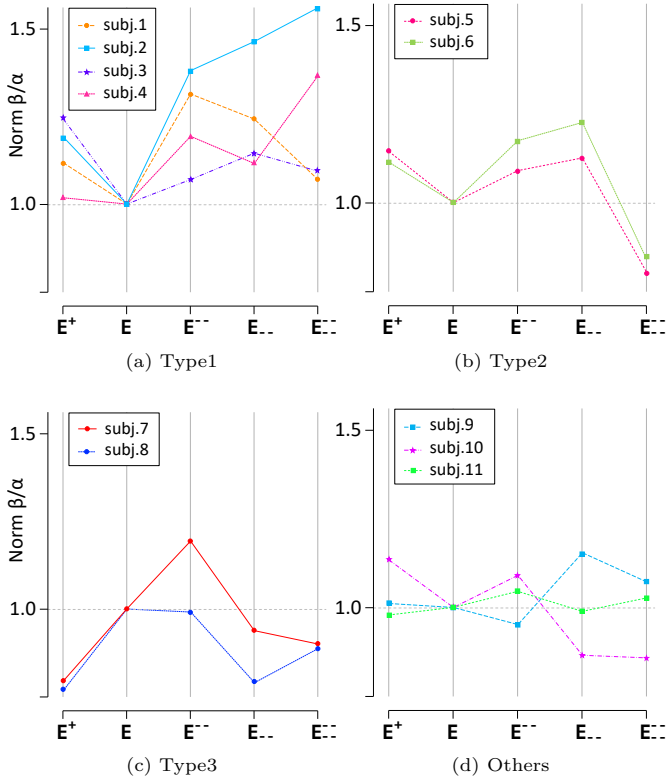


図5 各被験者のEに対する各環境の相対 β/α

認知負荷の指標である、 β/α を表す。4.1で述べた通り、 β/α は個人差による影響を最小化するため、各被験者の平均値で正規化している。箱ひげ図上部の線と値は検定の対象と、その検定の結果を表している。検定は有意水準を0.05としてウィルコクソン順位和検定で行った。例えば、EとE..の検定結果は0.23となっており、これは有意水準0.05を下回っていないため、EとE..には有意差がないと言える。

ウィルコクソン順位和検定による検定では、どの環境の認知負荷にも差は確認できなかった。特に、E..はランダムハイライトと筆記体のフォントでプログラムが表示される、今回の実験では最もプログラムを読むのに適していない環境であったが、それでもEとの比較に有意差は確認できなかった。しかし、E..とEそれぞれの中央値に注目すると、E..の方が β/α の値が小さく、発生した認知負荷がより小さかったと考えられる。続いて、プログラム読解に最も適した環境である、E+に注目すると、その中央値は他のいずれの環境の中央値よりもわずかに高くなっている。これらの結果は、プログラム読解に適していない環境でプログラムを読むほど、認知負荷が高くなるという我々の仮説と全く逆の結果を示している。従って、RQ1に対しては計測できないという結論となった。また、RQ2とRQ3はRQ1で認知負荷が計測できるという結果がある前提に作成されているRQであるため、これらのRQでハイライトやフォントがプログラム読解中の認知負荷に影響を及ぼすとは言い難いという結果となった。

5.2 各被験者個人の分析

次に、さらに詳しい分析をするため、個々の被験者に注目する。その分析結果として、各被験者の各環境での相対的な β/α

を図5に示す。この分析では、各環境の β/α の平均値をその被験者のEの平均値で正規化しているため、縦軸がEに対する相対 β/α となっている。そのため、どの被験者もEの値が1.0となっており、各環境の相対 β/α が1.0より高ければ、その被験者はその環境でのプログラム読解でより大きい認知負荷がかかっていたことを表し、その値が1.0より小さいければ、その逆で、より小さい認知負荷がかかっていたことを表す。また、この図では全ての被験者を4つのタイプに分類している。図5(a)のType1は全ての環境において、相対 β/α が1.0より高かった被験者のグループである。図5(b)のType2はE..以外の環境において、相対 β/α が1.0より高かった被験者のグループである。図5(c)のType3はE+において、相対 β/α が1.0より低かった被験者のグループである。図5(d)のOthersは上記3つのいずれのグループにも当てはまらない被験者のグループである。

Type1に属する被験者は、非プログラミング特化環境では認知負荷が大きくなるという我々の仮説に従う結果を示していた。特に、被験者2と3はE..やE-よりもプログラム読解に適していないE..の方が認知負荷が大きくなっていたことが分かる。しかし、Type1の被験者はEよりもプログラム読解に適しているE+でも他の環境と同様に相対 β/α が1.0より高くなっており、この点においては仮説に反する結果となっていた。続いて、Type2に属する被験者は、Type1の被験者と結果が似ているが、Type1の被験者の結果より仮説と異なる結果を示していた。Type2の被験者がType1の被験者と大きく異なっているのはE..であり、Type2の被験者は相対 β/α がE..で1.0より小さい値を示していた。次に、Type3に属する被験者のE+に注目すると、Eと比較して明らかに認知負荷が小さかったことが分かる。これはプログラミング特化環境でのプログラム読解ではより認知負荷が小さくなることを意味している。しかし、Type3の被験者は非プログラミング特化環境の多くで我々の仮説とは逆の結果を示していた。結論として、全ての環境において我々の仮説通りの結果を示していた被験者はいなかった。ただし、この分析は各被験者個人の結果に注目しており、カウンターバランスの効果がなくなってしまうため、この結果の妥当性は慎重に検討する必要がある。

6. 考察

全体として環境ごとの認知負荷の差を確認することはできなかった。その理由を、「脳波で認知負荷を計測するのは難しかった」、「内容の理解より実際の出力の計算に負荷がかかるプログラムだった」、「プログラムが高難易度すぎた」、「プログラム毎の難易度のばらつきが大きかった」の4つに大別して考察する。

6.1 脳波による認知負荷の計測が難しかった

1つ目の理由は、「脳波で認知負荷を計測するのは難しかった」可能性があるということである。この可能性に対して、認知負荷を脳波で計測し、その計測が可能であったことを示している研究がある[5]。さらに、我々は実験後、被験者に実験に関するアンケートを行ったが、多くの被験者は非プログラミング特化環境でのプログラム読解の方が困難であったと回答していた。それにもかかわらず、本稿の実験では仮説に反する結果が得られたとい

う事実から、環境に起因する認知負荷は脳波計測によって計測できない程度の小さい影響しか及ぼさないのではないかと考えた。本稿の実験では脳波への環境による影響は小さいという可能性を考慮して、非現実的なハイライトとフォントを設定したが、脳波への影響は得られなかった。よって、脳波への影響を大きくするためには、プログラムを読むことがさらに困難になるような環境を追加するのがよいのではないかと考えられる。例えば、1文字ずつ色が変わるようなさらに細かいハイライトにする、フォントをトークン毎に変更するなどの例が挙げられる。

6.2 視認より脳内での計算に負荷がかかった

2つ目の理由は、「内容の理解より実際の出力の計算に負荷がかかるプログラムだった」可能性があるということである。具体的な入力が与えられているプログラムの出力を考えるという作業には2つの段階が存在する。最初の段階はプログラムを確認し、プログラムの内容を把握する段階であり、ハイライトやフォントがプログラム理解に影響するのはこの段階である。その次の段階は入力に与えられている値を実際にプログラムの処理に従って計算する段階である。この段階では前の段階で理解したプログラムの処理に従っていくだけであるため、この段階にはハイライトやフォントによる影響がほとんどないのではないかと考えられる。これら2つの段階にかかる負荷の割合はプログラムによって異なっているが、今回の実験で実際に使用したプログラムは後者の段階にかかる負荷が大きいプログラムが多かった。そのため、今回の実験においてはプログラムを理解する際に、被験者がディスプレイに視線を向けていたとしても、その意識はディスプレイ上のプログラムには向いていなかったのではないかと考えられる。このような問題に対する解決策としては、後者の段階にかかる負荷を抑えるため、実験タスクをプログラムの出力を答えるのではなく、処理の内容を答えさせる形にするのが効果的であると考えられる。

6.3 難易度が高次であった

3つ目の理由は、「プログラムが高難易度すぎた」可能性があるということである。本稿の実験で使用したプログラムはプログラミングコンテストの問題を改題して作成されている。そのため、実験に使用するプログラムとしては難易度が高く、常に認知負荷がかかることによって環境に起因する影響が薄められてしまった可能性がある。この問題の発生を防ぐためには、本稿で使用したプログラムよりも難易度が低いプログラムを使用することが効果的ではないかと考えられる。

6.4 難易度が不均一であった

最後に、4つ目の理由は、「プログラム毎の難易度のばらつきが大きかった」可能性があるということである。今回の実験に使用したプログラムは全て同一のプログラミングコンテストの問題を参考にして作成したが、同一のプログラミングコンテストの同一ランクの問題でも開催回によって難易度が異なっていた可能性がある。プログラムの難易度はすべて等しいことが理想的ではあるが、ソフトウェア工学の世界ではプログラムの難易度の計測、及び統一化が難しいことはよく知られている。そこで、この問題の解決策としては、2, 3個の変数と、1つの分岐程度の分量である、論文内のプログラム例のような問題を採用すること

が有効であると考えられる。

7. おわりに

本稿では、プログラム読解中の認知負荷を脳波を使用して計測する実験を行った。しかし、いずれの環境間にも有意な差を得ることはできなかった。そのため、本稿では結果に有意差が得られなかった理由を考察した。

今後の展望としては、考察の改善案に基づく改善を行ったプログラムを使用し、再実験を行うことが挙げられる。さらに、ハイライトとフォントに起因するその他の影響を調査するため、新たなメトリクスとして理解の正しさや理解の速度の採用も有効であると考えられる。また、視線計測のようなその他の生体メトリクスの併用により、プログラム理解への新たな指標の提供が期待できるのではないかと考える。

謝辞 被験者実験に協力して頂いた方々に感謝する。本研究の一部は、日本学術振興会科学研究費補助金基盤研究(B)(課題番号:18H03222)の助成を得て行われた。

文 献

- [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, pp.1-5, 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, pp.396-407, 2019.
- [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, pp.350-355, 2016.
- [4] T. Ishida and H. Uwano, "Synchronized Analysis of Eye Movement and EEG during Program Comprehension," In Proc. International Workshop on Eye Movements in Programming, pp.26-32, 2019.
- [5] P.D. Antonenko and D.S. Niederhauser, "The influence of leads on cognitive load and learning in a hypertext environment," Computers in Human Behavior, vol.26, no.2, pp.140-150, 2010.
- [6] M.K.C. Yeh, D. Gopstein, Y. Yan, and Y. Zhuang, "Detecting and comparing brain activity in short program comprehension using EEG," In Proc. IEEE Frontiers in Education Conference, pp.1-5, 2017.
- [7] A. Sarkar, "The impact of syntax colouring on program comprehension," In Proc. Psychology of Programming Interest Group, p.8, 2015.
- [8] C. Hannebauer, M. Hesenius, and V. Gruhn, "[Journal First] Does Syntax Highlighting Help Programming Novices?," In Proc. International Conference on Software Engineering, pp.704-704, 2018.
- [9] T.R. Beelders and J.P.L. duPlessis, "Syntax highlighting as an influencing factor when reading and comprehending source code," Journal of Eye Movement Research, vol.9, no.1, pp.2207-2219, 2016.
- [10] F. Hirai, K. Yoshida, and I. Miyaji, "Comparison analysis of the thought and the memory at the learning time by the simple electroencephalograph," Multimedia, Distributed, Cooperative, and Mobile Symposium, pp.1441-1446, 2013.