

# 組み込みシステム開発のデバッグにおける熟練度による視線動作の特徴分析

馬場 建

Takeru BABA

## 1 はじめに

業務効率改善や家電製品の利便性向上などを目的に IoT 市場規模は拡大しており, IoT 技術者の需要が増加すると予想されている. IoT の開発は, 組み込みシステム開発を基礎技術として, ネットワークやセンシングなどの応用技術が必要であり, 技術の習得が容易でない. IoT の基礎技術である組み込みシステム開発の効率化は, 今後のソフトウェア工学の課題になると考えられる.

組み込みシステム開発は, 一般的なソフトウェア (以降, SW) 開発に加えて, マイコンや回路などのハードウェア (以降, HW) の要素が増え, システムの誤りは SW と HW の両方に発生しうる. 多くの場合, SW と HW の誤りは見分けがつかず, システムの誤り箇所を推定し, 修正することは初学者にとって容易でない.

SW 開発に関する先行研究では, 熟練者のデバッグ作業における特徴を明確化することで, デバッグ効率向上のための教材にできることが報告されている<sup>1)</sup>. したがって本研究では, 組み込みシステム開発特有のデバッグ作業における視線動作を分析し, 熟練度によるデバッグ過程の違いを定量的に明らかにすることを目的とする.

## 2 組み込みシステム開発に関する視線分析実験

### 2.1 実験目的

本実験の目的は, 組み込みシステム開発における視線動作の数学的モデルを生成することで, 熟練度の違いにより生じるデバッグ作業の特徴を明らかにすることである. また, 熟練者のデバッグ作業の特徴をもとに, 組み込みシステム開発における効率的な修正方針を明らかにする.

### 2.2 実験条件

被験者は, 組み込みシステム開発経験がある大学生の 7 名である. 開発経験の差により表れる特徴を明らかにするため, 経験年数をもとに被験者の熟練度を分類する. 組み込みシステム開発経験が 1 年以上である 3 名を「熟練者」, 1 年未満である 4 名を「初学者」と定義する. 実験のために Fig. 1 のような環境を構築した. 構築した組み込みシステム開発の環境の要素を下記に述べる.

- ディスプレイ: ソースコードと文字出力
- 問題用紙: 回路や関数, コンパイル方法などを記述
- Arduino: 電圧を制御するマイコン
- 回路: デジタル入出力に関する回路

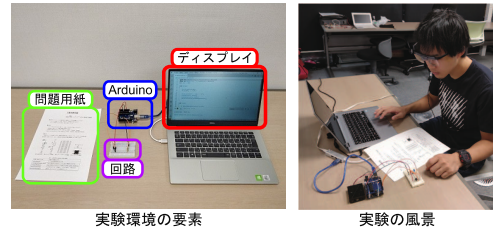


Fig. 1 実験環境の要素と実験の風景

被験者に与える課題は, スイッチの押下時に一定間隔で LED が点滅し, 押下時以外は LED が消灯するシステムの構築である. 被験者に課題として与えるシステムには, ディスプレイと回路に 2 つずつ誤りが含まれている.

### 2.3 実験手法

熟練度の違いにより生じるデバッグ作業の特徴を分析するため, 被験者に組み込みシステムのデバッグを行なう課題を与えた. 実験開始時には, システムの完成形を見せながら課題の説明を行なった. 説明の後, 被験者はシステムのデバッグ作業を行なった. 実験中に取得する情報は, アイトラッカーによる視線情報, ソースコード編集履歴などである. 実験の終了後, 組み込みシステム開発の経験歴に関するアンケートを実施した.

### 2.4 実験結果と考察

#### 2.4.1 実験時間の全体的な傾向に着目した分析

実験により取得したデバッグ中の視線情報から, 実験全体の視線の停留ごとの注目物体の推移を分析した. その結果, 熟練者は序盤に回路を中心に注目し, 初学者は序盤に回路とディスプレイ間の注目の遷移を繰り返す傾向があった. 熟練度ごとに遷移の傾向が異なったのは, 初学者は誤り箇所を推測できず, 修正方針が定まらない状態に陥っていたことが原因だと考えられる. したがって, 初学者に修正方針を教示することで, デバッグ効率が向上する可能性があるといえる.

#### 2.4.2 実験時間の部分的な傾向に着目した分析

取得した視線情報からデバッグ効率の差を引き起こす要因を分析するために, 注目物体の時間的推移において熟練度ごとにデバッグ方針に関する特徴があるかを調査した. 実験結果を視線の停留回数で 3 等分割することで, デバッグ箇所の偏りが存在するか検証した.

熟練者と初学者の, 注目物体の遷移に関するマルコフモデルを 3 つの段階で比較すると, 終盤のモデルに顕著

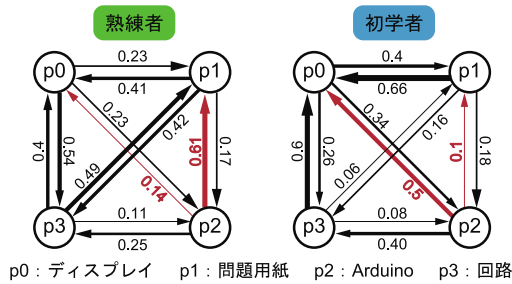


Fig. 2 実験の終盤における注目物体のマルコフモデル

な違いがみられた。終盤における注目物体のマルコフモデルを Fig. 2 に示す。Fig. 2 の矢印の太さは、遷移確率の高さを視覚的に表したものであり、矢印が太いほど確率が高い。各遷移確率に関して、Fig. 2 に示す赤色の矢印の遷移において統計的に有意な差があった。

問題用紙からディスプレイと回路に向かう遷移確率に着目すると、熟練者は両方の遷移確率が近い値を示したことから、熟練者は終盤にソースコードと回路を修正していると考えられる。また、ディスプレイと回路間の双方向の遷移確率が近い値を示したことから、熟練者は終盤にソースコードと回路に誤りがある可能性を等しく疑っていると推測できる。初学者は問題用紙からディスプレイへの遷移確率が高いことから、初学者は終盤にソースコードを修正していると考えられる。

#### 2.4.3 時系列データの数学的な分割

実験時間を数学的な手法で分割することで、熟練度ごとの修正方針を推定できるかを調査した。時系列データの分割をするために、GP-HSMM (Gaussian Process - Hidden Semi Markov Model) というガウス過程と隠れセミマルコフモデルを組み合わせた手法を用いた<sup>2)</sup>。

実験時間について5回ずつ分割を試行した際の、熟練者の結果を Fig. 3、初学者の結果を Fig. 4 に示す。図の上側は、視線の停留ごとに被験者が注目した物体の推移を示したグラフである。下側は、注目物体推移を GP-HSMM により分割した結果であり、熟練者は Fig. 3 下側の3クラス、初学者は Fig. 4 下側の4クラスに表れた。各クラスの意味を注目物体のマルコフモデルから調べると、熟練者は緑が「回路と問題用紙」に注目、赤が「ディスプレイと問題用紙と回路」に注目、青が「ディスプレイ」に注目する状態だった。初学者は緑と青が熟練者と同じ状態で、ピンクが「ディスプレイと問題用紙」に注目、黄が「回路と Arduino」に注目する状態だった。

2.4.2 項の結果を踏まえると、熟練者は序盤に回路、中盤にソースコード、終盤にシステムの動作確認という3つのデバッグの段階があることがわかった。それに対し初学者は、修正箇所をクラスをまとめると緑と黄が回路、赤と青がソースコードを修正している状態であり、初学者は前半に回路、後半にソースコードを修正するという

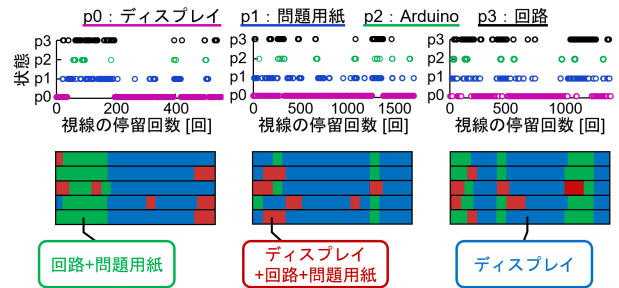


Fig. 3 熟練者における時系列データの分割結果

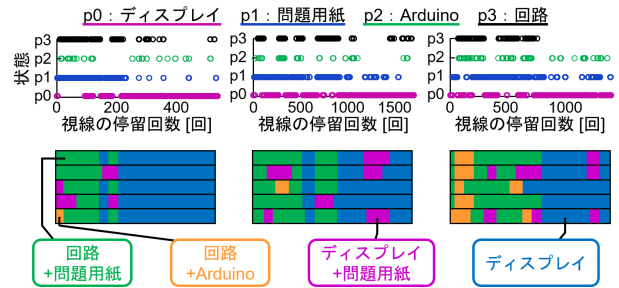


Fig. 4 初学者における時系列データの分割結果

2つのデバッグの段階があることがわかった。

### 3 結論

本研究では、組込みシステム開発のデバッグにおける視線動作を取得し、熟練度の違いが視線動作に与える影響を分析した。システムの動作確認をする終盤において、熟練者はソースコードと回路に誤りがある可能性を等しく疑うことがわかった。また、実験時間を分割することで、デバッグ作業をいくつかの段階に分けられることがわかった。時系列データの数学的な分割手法により、熟練者は3段階、初学者は2段階でデバッグを進めることがわかった。熟練者のデバッグ段階を参考にすることで、初学者がデバッグの方針を立てることを容易にし、デバッグ効率向上に繋がれると考えられる。

熟練者の動作確認方法を参考にすることで、教育現場における教師のアドバイス内容に応用できると考えられる。例えば、SWのみをデバッグの進展なく見続ける初学者に対して、HWも動作確認するように教師が誘導することで、システムの動作確認方法と組込みシステム開発のSWとHWの両方がシステムの挙動に影響するという特徴を教示できる。

### 参考文献

- 1) 應治 沙織, 上野 秀剛: コードレビュー時の読み方教示によるレビュー効果の変化, 情報処理学会研究報告書, Vol.2014-SE-185, pp.1-8 (2014).
- 2) 中村 友昭, Attamimi Muhammad, 長井 隆行, 持橋 大地, 小林 一郎, 麻生 英樹, 金子 正秀: ガウス過程の隠れセミマルコフモデルに基づく身体動作の分節化, 人工知能学会全国大会論文集, Vol.JSAI2016, p. 1035 (2016).