

プログラミング行動の測定と分析に関する一考察

大場みち子^{†1} 山口琢^{†2}

概要: ビッグデータ分析の対象は、ホワイトカラーの業務や学生の学習活動などの知的行動に及んできた。本研究は分析の対象としてシステム開発での操作行為に着目している。機能性や操作性に重きをおいた従来とは異なり、開発行為の測定性に重きをおく。作文の操作行為については、概念実証を経て実際のライティング講義に適用・評価する段階に進んでいる。他方で、実システムを開発するリアル PBL(Project Based Learning)のノウハウを蓄積してきた。本研究ではシステム開発 PBL でのプログラミングを対象とする。プログラミングスキルを獲得する上での成長や阻害の要因との関係が分析できれば効果的な指導の介入ができ、効率的なスキル獲得が期待できる。我々はプログラミング行動を把握し、教育的指導の介入を可能とするシステムの開発を目指してプログラミング行動を測定する操作モデルを提案し、プロトタイプシステムを開発している。本稿ではプロトタイプシステムを利用したプログラミング行動の実験結果とその分析手法の検討状況について報告する。

キーワード: プログラミング行動, 測定, PBL, 操作モデル

A Study on Measurement and Analysis of Programming Activity

MICHIKO OBA^{†1} TAKU YAMAGUCHI^{†2}

Abstract: The subjects of the Big Data analysis have covered intellectual actions such as white collar work and student learning activities. This research focuses on the operation behavior in system development as the object of analysis. Unlike in the past, which emphasizes functionality and operability, we place emphasis on the measurement performance of development activities. As for the act of writing composition, we are proceeding to the stage of applying and evaluating the actual writing lecture after proof of concept. On the other hand, we have accumulated know-how of real PBL (Project Based Learning) developing systems with actual users. Therefore, in this research, the target is system development, always new entrants, real PBL where the actual user is located will be the place of evaluation. Various skills are required for system development in real PBL. If we can analyze relationships with factors of growth and inhibition in acquiring these skills, effective teaching can be interposed and efficient skill acquisition can be expected. We will grasp the programming behavior and aim to develop a system that enables intervention of educational guidance. In this paper, we propose an operation model to measure programming behaviors for system development in real PBL and demonstrate the concept by prototype.

Keywords: Programming Activity, Project Based Learning, Measure, Activity Model

1. はじめに

ビッグデータ分析の対象は、ホワイトカラーの業務や学生の学習活動など知的活動に及んできた。システム開発、作文、読書など様々な知的活動のデジタル化と、ツールのクロスプラットフォーム化や国際化が進んだ。これにより、さまざまな知的活動の測定・分析を、実験室だけでなく、さまざまな実践の場面を対象にして、より細かい粒度 (granularity) で行えるようになってきた。電子書籍リーダーの機能で、「数多くの読者がハイライトした箇所を表示」、「読み終わるまでの平均時間表示」などは、このような測定・分析のうちでも商業化され広く知られた例である。

本研究では知的分析の対象としてシステム開発での操作行為に着目している。機能性や操作性に重きをおいた従来とは異なり、開発行為の測定性に重きをおく。作文の操作行為の分析[1][2]については、概念実証を経て実際のライテ

ィング講義に適用・評価する段階に進んでいる[3][4]。他方で、実システムを開発するリアルPBL(Project Based Learning)のノウハウを蓄積してきた[5]。PBL (Project Based Learning) など実践型ICT 教育でも、利用するICT ツールは課題に応じて学生の自由に任せられ、かつ利用する場所も学内にとどまらず学外も含めてさまざまである。このようなICT ツール利用の実態に対応すれば、システム開発に必要なとされる実践的な各種スキル獲得上の成長・阻害要因、スキル間の関係の分析、スキルの評価、および開発過程(学習)への教育的な介入を実現できると期待できる。

我々はプログラミング行動を把握し、教育的指導の介入を可能とするシステムの開発を目指してプログラミング行動を測定する操作モデルを提案し、プロトタイプシステムを開発している[11]。

本稿ではプロトタイプシステムを利用したプログラミング行動の実験結果とその分析手法の検討状況について報告する。また、実験から今後の課題を洗い出す。

2. 従来の研究・実践と課題

従来の研究や実践の課題は、次の2点に整理できる。マクロからマイクロへ- 測定・分析の細粒度化システム開発行動

*†1 *公立はこだて未来大学
Future University Hakodate
†2 フリーランス
Independent Researcher

の分析対象を、次のように大きく 2 種類に整理することができる。

- システム開発行動の結果・成果
- システム開発での試行錯誤など思考や操作行為

細粒度化とは、測定・分析の対象が、前者のみから後者を含むように広がることを意味する。実践型 ICT 教育では、例えばドキュメント作成およびプログラミングのスキルを測定・分析の対象とするとき、前者はプログラミング力と論理的文書作成力との関係の分析である。後者はプログラミングの思考過程の見える化の試みと分析、ドキュメントの作文操作行為とプログラミングの操作行為に着目した分析に該当する。

前者の方式による開発管理やノウハウ抽出・継承の取り組みでは、開発の規模や不良数などアウトプットの推移を分析したり(KPI, Key Performance Indicator), 既存の支援システムのログを分析したりするものであった[6][7][8]。この方式では、結果について評価することはできても、作業しつつある開発者に、きめ細かなタイミングでアドバイスすることができなかった。

実験室から実践

ICT ツールを使った活動の測定(観察)では、実践の場で使われる ICT ツールはさまざまであり、実験室のように「測定機能を実装した実験専用アプリ」を用意して統一することは非現実的である。また、そのようなアプローチには、アプリそのものの評価やアプリ間の比較に適用できないという原理的な困難もある。

例えば、特殊な測定ハードウェア装置を利用する方式は装置の配備 (deploy) にコストがかかるが、対照的に Web アプリケーションでは配備そのもののコストは比較的小さく、それによって測定がスケラブルである。別の観点では、ビデオ撮影などによる記録では、何をしているか分析可能なデータとするのに人手と時間がかかる。ルーブリックなどの工夫が研究・考案されているものの、統計処理対象のデータに既に人の解釈によるバイアスがかかっている。

専用の Web アプリであれば配備の点でスケラブルとなるが、さまざまな実践の場を測定しようとする、マルチプラットフォーム化や国際化といった開発コストが発生する。

人やモノの物理的な行動の測定では、ウェアラブルデバイスの低コスト化と普及・BYOD (Bring Your Own Device) 対応によって、スケラブル化かつ低バイアス化が実現されつつある。同様に、人の知的な行動に密着した(ウェアラブル) 機械的な(低バイアス) 測定方法の開発が必要である。

3. 目的とアプローチ

2 章で述べたことより、本研究の目的は、システム開発行動をマイクロに測定(観察・観測)・分析し、学習・指導・

改善に役立てる手法を開発することである。コーディング、プロジェクト管理などでの開発ツール操作に着目して、ツール操作のモデルを開発し、操作行動を測定する仕組みを設計・実装して評価する。

本研究では次のアプローチをとる:

- 測定用の特定のアプリを開発するのではなく、広く使われている多くのアプリケーションにアドオン可能な測定方式を開発し、
- プロトタイプを開発して、
- データを機械的に得て、
- それを分析できる手法を見出し、必要ならば新たに分析手法を開発して、
- 学習・指導に適用して効果をあげることで、当初の測定方式の有用性を示す。

4. ツール操作モデル

操作モデルは、図 1 のような簡単なモデルとした。すなわち、プログラミング行動とは、次のような内容で構成される「操作」の集合である。

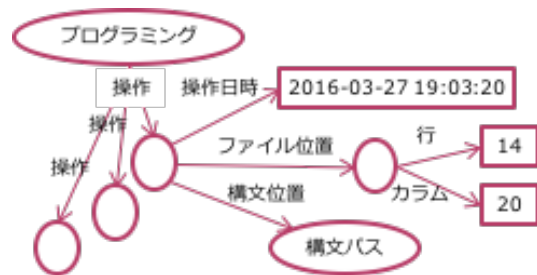


図 1 操作モデル

そして、それらはモジュール間のグローバルなイベント通信を監視することで記録できると期待できる。

- 操作した時刻
≡ イベント受信時刻
- 操作対象となった行・カラム(何文字目)
イベントの内容に含まれていると期待
- 操作対象のプログラム構文上のノード種別
≡ 操作の意味
イベントの内容に含まれている、または解析によって判明すると期待

5. 予備実験

予備実験では、簡単なツール操作モデルを設計して、それに基づいて測定するシステムのプロトタイプを開発して、測定データを取得することによって、3 章の目的とアプローチが実現可能かを確かめた。

イベント送信・受信の測定

具体的には、次の仮説(期待) を確かめることができた。近年の Web アプリは、オープンソースのモジュールを組

み合わせて開発されている。対話型のツール類は、ユーザ操作を最初のきっかけとして、グローバルオブジェクトへのイベント送信・受信によって疎な連携をしていると推察する(疎結合)。このイベントを受信することで、ユーザ操作に密着した測定が容易に実装できると期待できる。

題材は、JavaScript 言語でのプログラミングとした。3章のアプローチに基づいて、JavaScript 言語での開発環境を選定し、アドオンによってプログラミン活動の測定できるようにする。そして、プログラミング問題を解くという場面を想定し、実際に問題を解いてその活動を測定した。

Web ブラウザで JavaScript のプログラムを開発できるオープンソースの開発環境として、JS Bin[9] と Plunker[10] が知られている。これらはオープンソースでもある。今回は JS Bin を採用した。

測定のプロトタイプと得られた知見

開発環境は、複数のオープンソースのモジュールを組み合わせ合わせて使っていて、関数の実行と戻り値およびグローバルオブジェクトへのイベント発行・受信によって連携させることで実現されていることが確かめられた。使っているライブラリは、codeMirror(エディタ、構文ハイライト) と JS Hint(構文チェック)である。

なお、開発環境は構文チェックして問題箇所を画面表示する機能を持っているが、エディタモジュールと構文チェックモジュールとは、プログラムコード全体を渡すことで連携していて、行・カラムや構文位置を共有していない。

ライブラリ間のイベント送信を捉えて、時刻と合わせて記録することでプログラミング行動を記録できた。これはオープンソースの対話型開発環境が共通にもっている性質と言える。つまり、Web ブラウザで利用する、オープンソースのプログラミング環境の一つについて、それが複数のオープンソースのモジュールを組み合わせ合わせて実装されている。モジュール間はグローバルなイベントによって連携する疎結合になっていて、そのイベントを捉えることでプログラミング操作を記録できることが分かった。これは他の開発行動の測定にも同様の実装アプローチで実現できる見通しを得た。

6. 実験

被験者 1 名に対して実験を実施し、その操作ログを元に分析を実施した。実験の課題は図 2 の【課題】に示すとおりで、図 3 のプログラム中の「ここを修正」ではさまれた部分を修正して、正しい結果が表示されるように JavaScript プログラムを修正する問題を解くものとした。

JavaScript プログラムを編集したり、予約語をハイライトしたり、JavaScript を実行する仕組みは JS Bin が提供するものであり、手を加えていない。

【課題】

A と B はどちらも、次のような値を返す関数です。

A と B の返す値が異なります。A を修正して、同じになるようにしなさい。

A と B が返す値:

n=0 のとき a を返す

n=1 のとき 1 を返す

n>1 のとき、次のように定義される数列の n 項を返す

$$A(0)=a, A(1)=1, A(n)=A(n-1)+A(n-2)$$

ただし、結果が int の範囲を超える場合は数学的に正しい値を返せなくても良い

B は関数の再帰呼出しを使って計算しています。

図 2 課題

```
var i = 0,
    k = 0,
    l = 0,
    body = document.getElementsByTagName('body')[0],
    txt = "",
    div = {};

/* ↓ここを修正 */
function A(a, n) {
    var j = 0,
        x1 = 0,
        x2 = 0,
        x3 = 0;
    if (n === 0) {
        return a;
    } else if (n === 1) {
        return 1;
    } else {
        x1 = 1;
        x2 = a;
        for (j = 2; j <= n; j++) {
            x3 = x1 + x2;
            x1 = x3;
            x2 = x1;
        }
        return x3;
    }
}

/* ↑ここを修正 */

function B(a, n) {
    var j = a;
```

```

    if (n === 0) {
        return j;
    } else if (n == 1) {
        return 1;
    } else {
        return B(a, n - 1) + B(a, n - 2);
    }
}

for (i = 0; i < 10; i++) {
    k = A(3, i);
    l = B(3, i);
    txt = k + " == " + l + " " + (k == l);
    div = document.createElement('div');
    div.textContent = txt;
    body.appendChild(div);
    // $("body").append("<div>" + k + " == " + l + " " + (k == l)
    + "</div>");
}

```

図3 JavaScript プログラム

測定データ, 編集操作ログ

図4は、記録されたプログラミング操作ログを表示したものである。図4のデータを元に編集対象にしたプログラム行と時間の関係を散布図にしたものが図5である。同様に、編集対象にした構文要素の種類と時間との関係を散布図にしたものが図6である。ここで、構文要素とは図7に示すようにプログラムの各構文要素に名称を付けたものである。図7の1行目のif文の条件式は”if-test”という構文要素であり、2行目のfor文の構文要素は初期化式、条件式、変化式、実行文から成り、それぞれ”for-init”, ”for-test”, ”for-update”, ”for-body”と名付けている。

1	日時	行	文字	コード全体
	2017/3/11 00:04:35	21	24	<pre> var i = 0; k = 0; l = 0; body = document.getElementsByTagName("body")[0]; txt = ""; div = {}; // ここを修正*/ function A(a, n) { var j = 0; x1 = 0; x2 = 0; x3 = 0; if (n === 0) { return a; } else if (n == 1) { return 1; } else { x1 = 1; x2 = a; for (j = 2; j <= n; j++) { x3 = x1 + x2; x1 = x3; x2 = x1; } } } </pre>
2				

図4 プログラミングの操作ログ

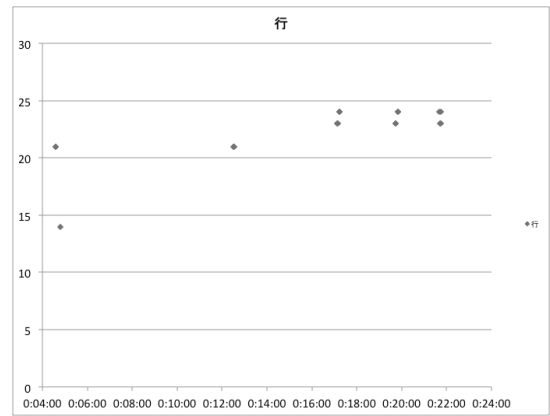


図5 編集の対象にした行に着目した散布図

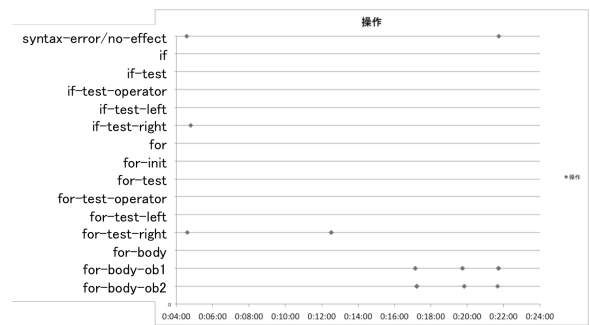


図6 編集の対象にした構文の種類に着目した散布図

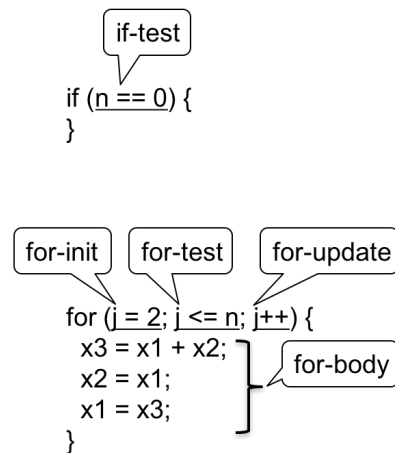


図7 構文要素の例

結果

- 編集の対象とした行に着目した場合の結果は、図5よりつぎの通りである。
 - 当初(0:04:00(4分)~0:13:00), 15行~21行を修正していた。続いて(0:17:00頃)から23行~24行を修正し始めた。その後、正解に達した。
 - 全体が18分。15行~21行の探索に13分(72%), 23行~24行の探索に5分(28%)かけた。
- 編集の対象とした構文要素に着目した場合の結果、図6より次のとおりである。

- 当初(0:04:00(4分)~0:13:00), if-test や for-test など判定部分を修正していた. 続いて(0:17:00 頃から)for文の本体を修正し始めた. その後, 正解に達した.
- 全体が 18 分. if 文や for 文の判定部分の探索に 13 分(72%), for 文本体の探索に 5 分(28%)かけた.

考察

行に着目した結果からは, 時間経過のなかでどのあたりの行を操作しているのかわかる.

構文要素に着目した結果からは, 回答者(被験者)が次のように考えたのではないかと, 類推できる.

まず, 判定部分に着目した. 続いて, for 文の本体に着目した.

以上より, 要素に着目して記録・分析する方が, 行・カラム(行内の文字位置)に着目するよりも, 有益ではないかといえる.

7. 今後の課題

今後の課題としてつぎのようなことが言える.

直近の課題として, より多くの被験者や演習問題を使った実験をしていく必要がある. 現システムは Proof Of Concept のための最小限の実装にとどまっているが測定・分析システムを開発・改善していく. また, どのような分析ができるかや, どういう演習問題を与えれば, 次につながる分析ができるかの検討も必要である.

近い将来の課題として, 記録・分析対象とする操作の粒度はこれでよいか? 例えば操作回数を問題にするのであれば, 同じ場所を連続して修正するなら, まとめて1回の操作とみなすのか. 分析にかける前に, どのようにデータを補正する必要があるか?

その場合, 同じ場所とはどの範囲なのかも検討する必要がある. たとえば, つぎのようなことが考えられる.

- for-test 内なら, どこを編集しても同じ場所
- for-test 内でも, for-test-operator, for-test-left, for-test-right は別の場所

- どこまで細かく見れば役に立つのか?

遠い将来の課題として, 次に述べる「期待する結果」のようなことが分かったら, 指導方法の改善に役立つか? や演習問題の作り方の改善に役立つか? が挙げられる.

近い将来, 期待する結果としてつぎのことが考えられる. 被験者や演習問題の種類を増やせば, 分かりそうなことはつぎの点である.

(1)着目される場所の傾向

- 着目されることが多い/少ない, 構文上の場所: if 文の判定, for 文の判定など

- 最初に着目する場所の傾向

(2)所要時間/所要操作の内訳

- 最初に着目した場所が間違っている場合, 別の場所に着目するまでの時間/操作回数. これは, 誤りであると気づく

までの所要時間または所要操作回数とみなせると考えられる.

- 正しい場所に着目してから, 正解にいたるまでの所要時間/所要操作回数など

遠い将来, 期待する結果としてつぎのことが考えられる.

開発実績(システム開発リポジトリ)の分析と照らし合わせて①開発者が着目しやすい場所と, バグを作り込みやすい場所は一致しているのか? ②指導・経験によって一致するように変化するのか? この変化を開発者の成長とみなしてよいか? などである.

8. おわりに

プログラミング行動を把握し, 教育的指導の介在を可能とするシステムの開発を目指してプログラミング行動を測定する操作モデルを提案し, プロトタイプシステムを開発している. 今回, プロトタイプシステムにより実験をした結果, 編集した構文要素に着目して記録・分析する方が行やカラムに着目するよりも有益であることが分かった. 実験の結果を踏まえて, 直近, 近い将来, 遠い将来の課題を洗い出すとともに, 近い将来および遠い将来に期待する結果を検討した.

今後は今回検討した課題について解決を図っていく.

謝辞 本研究は JSPS 科研費 16K04798 の助成を受けたものである.

参考文献

- [1]山口琢, 大場みち子, 高橋慈子, 小林龍生, 高橋修, 編集操作の測定でアプローチする自然後処理の提案, 情報処理学会研究報告デジタルドキュメント (DD), DD-097-03, 2015.
- [2]山口琢, 高橋慈子, 小林龍生, 大場みち子, 作文行動の測定と分析:大学生と社会人の比較例, 研究報告教育学習支援情報システム (CLE), 2015-CLE-17(29), (2015.11).
- [3]高橋慈子, 山口琢, 大場みち子, 小林龍生, 文書作成教育におけるトピックライティングツール活用と効果, 情報処理学会研究報告 DC, [ドキュメントコミュニケーション] 2016-DC-101(9), (2016.03).
- [4]大場みち子, 山口琢, 高橋慈子, 小林龍生, 文章作成における論理的文章評価と編集操作との関係分析, 研究報告コンピュータと教育 (CE), 2016-CE-137(6), 1-8 (2016-11-26).
- [5]大場みち子, 伊藤恵, 情報教育シンポジウム 2014 論文集 2014(2), 81-88, 2014-08-17.
- [6]高田義広, 鳥居宏次 "プログラマのデバッグ能力をキーストロークから測定する方法". 電子情報通信学会論文誌. D-I, J77-D-1(9): 646-655 (1994-09)
- [7]吉村巧朗, 亀井靖高, 上野秀剛, 門田暁人, 松本健一: "ブレークポイント使用履歴に基づくデバッグ行動の分析(デバッグ, モデル駆動開発)". 電子情報通信学会技術研究報告. KBSE, 知能ソフトウェア工学 109(307): 85-90 (2009-11).
- [8]秀毛 嶺維馬, 奥野 拓プログラム理解支援を目的とした分散ペアプログラミングのコミュニケーションログの活用, 電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス 113(269), 117-121, 2013-10-17.
- [9]JS Bin - Collaborative JavaScript Debugging <https://jsbin.com> [Accessed: 2017.02.15].

- [10]Plunker <https://plnkr.co/> [Accessed: 2017.02.15].
- [11]大場みち子, 山口琢, リアル PBL におけるプログラミング行動を測定するシステムの開発, 情報処理学会研究報告 CE, [コンピュータと教育] 2017-CE-139(09), 2017.