

ソフトウェア開発行動の分析支援を目的とした映像検索システム

中島田 義敬, 森崎 修司, 門川 暁人, 松本 健一, 鳥居 宏次

奈良先端科学技術大学院大学 情報科学研究科

〒 630-01 奈良県 生駒市 高山町 8916 番地の 5

{yosita-n, shuuji-m, akito-m, matumoto, torii}@is.aist-nara.ac.jp

あらまし ソフトウェア開発行動の分析において、開発者の行動を記録したビデオ映像の中から分析対象とする特定の行動を発見、抽出することは必ずしも容易ではない。本研究では、ソフトウェア開発行動の分析支援を目的として、ビデオ映像から特定の行動が写っている場面を効率よく検索するためのシステムについて述べる。このシステムでは、キーストロークや注視点といった、開発行動に関わり、かつ、ビデオ映像と並行収集可能なデータ(行動履歴データ)をビデオ映像の検索に利用する。具体的には、行動履歴データの時間変化をグラフ表示し、データの値や時間変化のパターンに基づいて、特定の開発行動の発生時刻を同定することが可能である。プログラムのデバッグ作業が記録されたビデオ映像にシステムを適用した結果、「次に成すべき作業が分らずに悩んでいる」という開発行動の発生を8カ所中6カ所まで容易に検索することが出来た。

キーワード ソフトウェア工学実験, 開発プロセス分析, 行動履歴データ, 映像検索, 分析システム

A Video Recordings Retrieval System Supporting Analysis of Software Development Processes

Yoshitaka NAKASHIMADA, Shuuji MORISAKI, Akito MONDEN,
Ken-ichi MATSUMOTO, and Koji TORII

Graduate School of Information Science, Nara Institute of Science and Technology

8916-5 Takayama-Cho, Ikoma-Shi, Nara 630-01

{yosita-n, shuuji-m, akito-m, matumoto, torii}@is.aist-nara.ac.jp

Abstract In analyzing software development process, it is not easy to find or extract the subject's specific behavior. This paper proposes a retrieving system using behavior data (key stroke, gaze point on CRT, etc.) for video recordings of software development activities. This system shows graphs of process data, so that identifying characteristic behaviors and specify the time they occurred are easy. In order to evaluate this system, we conducted an experiment to retrieve subject's specific behavior such as "lost himself" from the behavior data. There were 8 scenes identified by having an interview with the subject. It resulted in 6 out of 8 scenes being automatically retrieved.

key words software engineering experiment, analysis of development process, behavior data, video recordings retrieval, analysis system

1 はじめに

ソフトウェアの開発効率や品質を向上させるなどの目的のために、ソフトウェア開発行動の分析が盛んに行われている。例えば、文献 [7][9] では、デバッグの効率に影響する要因を明らかにすることを目的としてデバッグ行動の分析を行っている。文献 [6] では、プログラミング作業中の知識獲得過程を明らかにすることを目的として、プログラミングの初心者が試行錯誤しながらプログラミングする過程の分析を試みている。文献 [3][5][13] では、プログラミングの熟練者と初心者のスキル差を明らかにすることを目的として、プログラムを理解するプロセスを分析している。

開発行動の分析においては、開発者の行った特定の行動に注目し、行動に至った動機や結果との因果関係の解明、行動のモデル化などが行われる。ただし、注目すべき行動が明らかでないような分析の初期段階においては、特定の行動に注目せずに行動全般を追跡することにより、特徴的な行動について改めて分析する場合がある。

しかし、開発行動の分析には多大な時間を要するという問題がある。開発者の行動はビデオ映像として記録される場合が多いが、長時間にわたって記録されたビデオ映像の中から、分析対象とする特定の行動が写っている場面を特定したり、特徴的な行動を発見することは必ずしも容易ではない。特定の行動が写っている場面を全て特定したり、特徴的な行動を発見するためには、ビデオ映像を繰り返し再生して見る必要があるが、文献 [7] で指摘されているように、実験に要した時間に対して、分析には 5~8 倍もの長い時間を要する。

本稿では、ソフトウェア開発行動の分析支援を目的として、開発行動が記録されているビデオ映像を、効率よく検索するためのシステムについて述べる。

本システムでは、キーストロークや注視点といった、開発行動に関わり、かつ、ビデオ映像と並行収集可能なデータ（行動履歴データと呼ぶ）をビデオ映像の検索に利用する [10]。具体的には、次の 2 つの機能を持つ。(1) 行動履歴データの時間変化をグラフ表示する機能。開発者が特徴的な行動を行った時刻を、行動履歴データから特定するのに有用であると考えられる。(2) データの値や時間変化のパターンに基づいて、特定の開発行動の発生時刻を検索し、その時刻の映像を再生する機能。特定の行動が写っている場面の映像をすべて見る場合に有用で

あると考えられる。

プログラミング作業の分析に本システムを適用した結果、分析者は、行動履歴データのグラフ表示を見ることにより、いくつかの特徴的な行動を発見することができた。また、「次に成すべき作業が分からずに悩んでいる」という開発行動の発生を 8カ所中 6カ所まで容易に検索することが出来た。このシステムを利用すれば、開発行動の分析に要する時間が大幅に短縮され、より詳細な分析や分析結果の多方面への応用が期待される。

関連研究としては、GUI の操作履歴の分析支援を目的とした文献 [1][8] などのシステムがあるが、本システムでは、ソフトウェア開発行動の分析を対象としている点が異なる。また、本研究と同じく行動履歴データを用いて開発行動の分析支援を行う方法が小林ら [4] によって提案されているが、被写体である人間の行動等をモデル化する必要があり、モデルを作るための分析等には必ずしも適用できないといった問題点がある。

以降、2 章では、ソフトウェア開発行動の分析の現状について検討し、システムにより支援すべき分析作業を明確にする。3 章では、本システムを用いて行う 2 通りの分析方法を提案する。4 章では、システムの構成と機能について述べる。5 章では、システムを用いて行った分析の具体例を示し、システムの有用性の評価を行った結果について述べる。6 章は、まとめと今後の課題である。

2 支援すべき分析作業

開発行動の分析においては、(1) 特徴的な行動の発見、および、(2) 特定の行動が発生した時刻の特定を行い、行動に至った動機や結果との因果関係の解明、および、行動のモデル化などが行われる。しかし、ビデオ映像を見ることによって (1)(2) を行うには多大な時間を要する。

特徴的な行動を発見する作業、および、特定の行動が現れた時刻を特定する作業を支援するためのシステムが必要とされる。

開発行動の分析の例としては、柳ら [12] は、バグを混入させる際の行動に限定してプログラミング行動を分析している。分析の結果、バグの混入に強く関連する行動パターンをいくつか発見している。その後、各行動パターンが現れた全ての場面を特定し、特徴が現れた回数や、特徴の出現とバグの混入の関係、特徴が現れた原因などを分析している。高田ら [11] は、プログラミング行動を 3 種類 (コン

パイル、プログラム実行、プログラム変更)に分類して分析し、各被験者ごとに、各行動が開始する時刻、回数、継続時間などを調査している。そして、プログラマ間に共通に現れる特徴的な行動パターンや、プログラマによって異なる特徴的な行動パターンをいくつか発見している。発見した特徴的な行動パターンを定量的に議論するためのプログラミング行動のモデルを定義している。門田ら [7]、吉川 [13] は、特定の行動に注目せずに、デバッグ作業者の行動や意図を追跡し、各試行の行動の共通点や相違点をいくつか発見している。その後、特定した行動について分析を深めている。

3 本システムによる開発行動の分析方法

本章では、本システムを用いて行う開発行動の分析方法を2通り紹介する。

まず、本研究において仮定する、開発者の行動と行動履歴データの対応について述べる。次に、本システムにおいて行動履歴データを用いることの利点について述べ、本システムを用いて行う2通りの分析方法について具体的に述べる。

3.1 開発者の行動と行動履歴データの対応

本研究では、図1のように、開発者の特定の行動、および、特定の状態変化は、行動履歴データの特定の値の変化に対応していると仮定している。

この仮定の下では、ビデオ映像を見ることにより特定できる開発者の任意の行動は、行動履歴データから特定できると考えられる。例えば、開発者がある特定のコマンドを入力しているという行動は、開発者のキーストロークから読みとることができる。

3.2 行動履歴データを用いる分析の利点

本研究では、行動履歴データを用いることにより、従来問題となっていたビデオ映像による分析には時間がかかるという欠点を補うことができると考えている。具体的には、ビデオ映像を見て行う分析と比べると、行動履歴データを見て行う分析は、次の2点において優位であると考えられる。

利点 1. データ全体を一度に見ることができるので、特徴的なデータの変化を容易に発見したり単時間で分析できると期待される。ビデオ映像は全体を一度に見ることができないが、行動履歴データの値をグラフなどにより表示すれば、データ全体を一度に見ることができるので、データ全体の傾向を掴んだり特徴的なデータの変化を容易に読みとることができる。

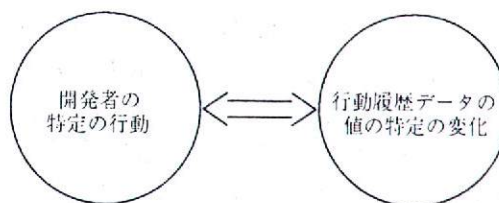


図 1: 開発者の行動と行動履歴データの対応

例えば、データ中に繰り返し現れる開発者の特徴的な行動パターンを容易に発見したり、複数の試行の行動履歴データを同時に見ることにより、試行間の共通点や相違点も容易に発見できる可能性がある。

利点 2. 行動履歴データに対して検索を行うことにより、特定の行動が現れた場面を容易に特定することができるので、単時間で分析できると期待できる。ビデオ映像だけから、特定の行動が写っている場面を検索することは容易ではないが、行動履歴データは数値データであるので検索を行うことが可能である。

例えば、何らかの特徴的な開発者の行動を発見した場合には、その行動が現れた回数を調べたり、その行動が写っている他の時刻の映像も分析する場合がある。ビデオ映像を見て、その行動が写っている場面を全て特定するには多大な時間を要する可能性があるが、行動履歴データの値や時間変化のパターンに基づいて、特定の開発行動の発生時刻を検索できれば、分析に要する時間が短縮できると期待される。

3.3 開発行動を分析する2通りの方法

本システムを用いて開発行動を分析する場合には、ビデオ映像と行動履歴データの両方を分析に利用することができる。

本節では、特定の行動が写っている映像だけを詳しく調べたい場合に有効な方法として、2通りの分析方法を紹介する。分析にはどちらの方法を用いてもよい、両方用いてもよい。

3.3.1 方法1 (行動履歴データから行動を特定する方法)

この方法は、特定の行動が現れた時刻を検索するための検索文を、行動履歴データだけから作成する方法である。

具体的な手順は、まず、行動履歴データのグラフ表示を見ることにより、グラフ中に繰り返し現れる

パターンを発見したり、複数の試行のグラフを同時に見ることにより、試行間で共通に現れるパターンを発見する。そして、発見したデータパターンを検索するための検索文を作成する。次に、作成した検索文を用いて検索を行い、検索により特定した時刻のビデオ映像を見て開発者の行動を詳しく調べる。

3.3.2 方法2 (ビデオ映像から行動を特定する方法)

この方法は、特定の行動が現れた時刻を検索するための検索文を、ビデオ映像を見た後で行動履歴データを見ることにより作成する方法である。

具体的な手順は、まず、ビデオ映像を見て、分析対象とする特定の行動が現れた時刻を全て特定する。次に、特定したそれぞれの時刻について、行動履歴データを見てデータの値や時間変化のパターンの共通点を発見し、発見した特定のデータパターンを検索するための検索文を作成しておく。別の試行で同じ分析を行う場合は、作成した検索文を用いて検索を行えばよいので、ビデオ映像を見る必要がなく、分析に要する時間を短縮できると期待できる。

4 映像検索システム

本システムを用いることにより、3章で述べた2通りの方法で、開発行動の分析を行うことができる。まず、本システムで用いる行動履歴データを収集する仕組みについて述べ、次に、システムの構成、機能について述べる。

4.1 データ収集システム

本節では、行動履歴データを収集するためのシステムについて述べる。本システムを使用するためには、あらかじめ開発者の行動を観測して、行動履歴データ、および、開発者の行動を記録したMPEG形式の映像データを収集しておく必要がある。

今回構築したシステムは、Ginger2システム[2]を用いて収集できるデータを扱うことにした。Ginger2システムとは、一人の被験者が計算機を使って行う作業について、データを収集、分析するシステムで、本システムもGinger2システムの一部に含まれる。Ginger2システムによるデータ収集環境は、図2のように2室に分割されている。1室は、被験者にソフトウェア開発を行ってもらう開発室である。もう1室は、実験者がデータを収集する観測室である。

Ginger2システムにより、現実のソフトウェア開発行動の観測実験で収集されているような様々な種類のデータを収集することができる。データの書式

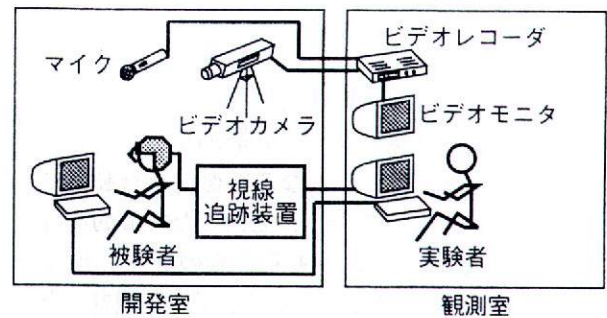


図2: Ginger2のデータ収集環境

(MON形式)は、文献[2]を参考にして頂きたい。

現状のGinger2システムにより収集可能な行動履歴データの種類の種類は次の通りである。

- 計算機画面の変化
 - ウィンドウ (kterm) の開閉, 位置, 大きさの変化
 - 各ウィンドウ内のテキストカーソルの移動
 - 各ウィンドウ内のテキストの変化
 - ウィンドウの重なりの変化
- 計算機に対する被験者の操作
 - マウスの移動
 - マウスボタンの操作
 - キーストローク
- 映像と音声
 - 被験者の挙動や表情
 - 発話
- その他
 - 被験者の注目している計算機画面上の点 (注視点) の遷移 (視線追跡装置によって収集される)
 - 被験者の手や頭や肩などの動き (3次元運動計測装置によって収集される)
 - 被験者の脈拍数の変化
 - 実験中の実験者が気づいたことをその場で残したメモ

4.2 システムの構成

本システムの構成を図3に示す。本システムは、グラフ表示部と検索部から構成される。

検索部では、分析者が入力した検索文を元に、行動履歴データに対して検索を行い、検索により特定した場面の映像を再生する。

グラフ表示部は、行動履歴データを読み込んで、グラフを分析者に呈示する。

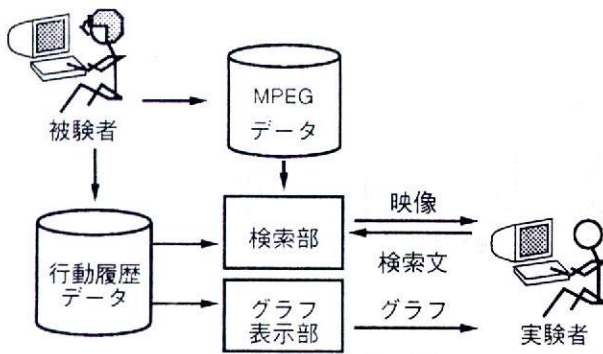


図 3: システムの構成

4.3 システムの機能

2章で述べた分析作業を支援するために、本システムは次の2種類の機能を持たせた。以下では、それぞれの機能の詳細について述べる。

4.3.1 データをグラフ表示する機能

グラフ表示部では、時間軸に対して値を持つデータをグラフ表示できる。データの例としては、単位時間あたりのキーストロークの頻度や、注視点の位置などである。状態遷移を表すデータも、各状態にそれぞれ異なる数値を割り当てることにより、グラフ表示が可能である。例えば、注視点が画面内にある状態を1とし、画面外にある状態を0とすることにより、注視点の遷移をグラフ表示することができる。

グラフを表示する機能は、さらに次の3つの機能に分けられる。

機能 1. データ全体を表示する機能である。全体の傾向を把握したり、特徴的なパターンのデータを発見するのに有効であると考えられる。

機能 2. データの任意の部分を拡大して表示する機能である。一部分だけを詳しく見たい場合に有効であると考えられる。

機能 3. 複数のデータを同じ時間軸上で同時に表示する機能である。異なる種類のデータ間の依存関係や、複数の試行で収集した同じ種類のデータ間の依存関係を調べるのに有効であると考えられる。

システムにより打鍵の頻度のデータ、および、注視点の位置のデータをグラフにして表示した例を図4に示す。図4には、4つのグラフが表示されているが、各データについて全体を表すグラフと、部分を拡大したグラフの2つが表示されている。全体のグラフ、および、部分を拡大したグラフは、それぞれ各データの時間軸が一致している。

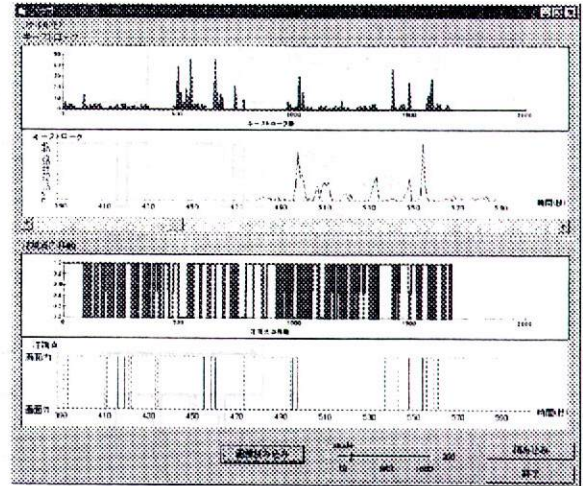


図 4: グラフの表示の例

4.3.2 特定の行動の発生時刻を検索する機能

検索部では、行動履歴データの値や時間変化のパターンに基づいて作成された検索文によって、特定の開発行動の発生時刻を検索を行う。複数のデータ間の依存関係を表すような検索文の作成も可能である。

検索部では、検索条件として、個々のデータ中に現れるパターンと、各データにおいて指定されたパターンの組合せの2つを指定することができる。前者としては、“行動履歴データの値がある条件を満たす状態が、ある一定時間続いた場面”が指定できる。後者としては、“複数のパターンが同時に現れた、あるいは、どちらか片方が現れた場面”が指定できる。例えば、“単位時間あたりの打鍵の回数が x_0 以上かつ x_1 以下である状態が t_0 秒以上継続し、かつ、注視点が CRT 画面内にある状態が t_1 秒以上継続した場面”は、次のように記述できる。

$$\{\text{key: } x_0 \leq x \leq x_1, t \geq t_0\} \text{ and } \{\text{eye: } x = 1, t \geq t_1\}$$

“key:” および “eye:” は、後に続く式が、キーストローク、および、注視点のデータに対する条件であることを示している。 x および t は、行動履歴データの値、および、時刻を表している。“ $x_0 \leq x \leq x_1$ ”、“ $t \geq t_1$ ”などの式は、 x および t に対する制約条件である。 $\{X\} \text{ and } \{Y\}$ は “XかつY” を表し、 $\{X\} \text{ or } \{Y\}$ は “XまたはY” を表す。

5 評価実験

本章では、構築したシステムを用いることにより、3章で述べた2通りの分析方法の実現可能性に

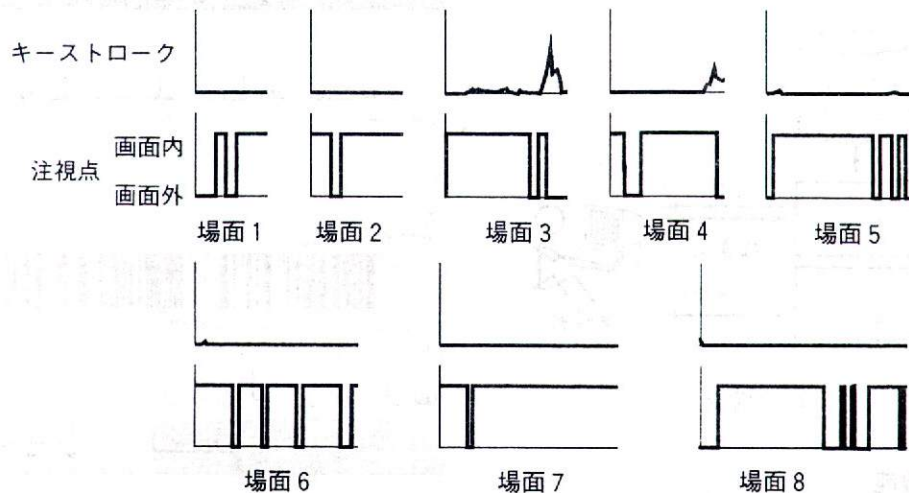


図 5: 悩んでいる場面のデータ

ついて評価する。また、実験の結果について考察することにより、3.1節で述べた仮定の妥当性についても検討する。

分析方法2においては、被験者の特定の行動が現れた場面をビデオから特定するが、評価実験においては、被験者が“次に成すべき作業が分からずに悩んでいる”という行動を特定することにする。初心者がどのような場面で悩むかを明らかにすることにより、初心者に対する効果的な教育方法を考えることができるかと期待される。

5.1 実験方法

2通りの分析を同時に行うために、“プログラミングの初心者が小規模なプログラムを作成する行動”を観測して、被験者の挙動を記録したMPEG映像、発話、キーストロークの履歴、被験者の見ている画面上の注視点の履歴の4つのデータを収集する。キーストロークと注視点データは、多くのソフトウェア開発行動の観測実験で、実際に収集されている ([7][13] など)。

実験においては、簡単なプログラム仕様書を用意し、1名の被験者(初心者のプログラマー)にC言語でプログラムを作ってもらい、プログラムの仕様は“標準入力を与えられた文字列を標準出力で返す”というごく単純なものである。

被験者は1名の大学院生である。この被験者は、プログラミングやC言語の知識はほとんどない。

5.2 実験の手順

実験の具体的な手順は以下の通りである。

STEP 1. 実験の準備 分析者は被験者に対して、C言語の基本的な文法や、よく使われる関数に

ついて、一通り教育する。

STEP 2. データの収集 被験者に仕様書を与え、プログラムの設計から、コーディング、テストまでを行ってもらい、机の上にはC言語の参考書、および、プログラムの仕様書を置き、自由に参照することを許した。被験者が考え込んでしまって、3分以上作業の進捗がみられない場合には、被験者にアドバイスを与える。

STEP 3. (方法1による分析) プログラムのテストが終了したら、分析者は、行動履歴データのグラフ表示を見て特徴的なパターンを探し、検索文を作成する。作成した検索文に基づいて検索を行い、検索によって特定した時刻の映像を再生することにより開発者の行動を詳しく調べる。

STEP 4. (方法2による分析) 分析者は、被験者の行動記録した映像を再生しながら被験者にインタビューを行い、悩んでいた場面の時刻を全て特定する。特定した時刻の行動履歴データのグラフを見て、各時刻に共通するパターンを発見し、検索文を作成する。作成した検索文に基づいて検索を行い、検索によって特定した時刻の映像を再生することにより開発者の行動を詳しく調べる。

5.3 結果

5.3.1 分析方法1を用いた結果

打鍵の頻度と注視点の位置のグラフを見ることにより、3つの特徴的なパターンを発見した。

パターン1. キーストロークの頻度が極端に多い部分があった。この部分を検索するために、“1

秒あたりの打鍵の回数が10回以上である”という場面を表す次のような検索文を作成した。

{key: $x \geq 10, t \geq 1$ }

検索した時刻の映像を再生すると、しきりにカーソルを移動させている行動が観測された。

パターン 2. 長時間打鍵が行われていない部分があった。この部分を検索するために、“1秒あたりの打鍵の回数が0である状態が30秒以上続く”という場面を表す次のような検索文を作成した。

{key: $x = 0, t \geq 30$ }

検索した時刻の映像を再生すると、プログラミングの本を読んでいる行動と、画面をじっと見ている行動の2通りが観測された。

パターン 3. 長時間注視点が計算機の画面外にある(被験者が画面を見ていない)場面があった。この部分を検索するために、“注視点が画面外にある状態が30秒以上続く”という場面を表す次のような検索文を作成した。

{eye: $x = 0, t \geq 30$ }

検索した時刻の映像を再生すると、プログラミングの参考書を読んでいる行動が観測された。

5.3.2 分析方法2を用いた結果

ビデオを見て、被験者にインタビューして、被験者が悩んでいた場面を8箇所特定した。特定した場面の打鍵の頻度、および、注視点のデータのグラフを、図5に示す。

図5より、被験者が悩んでいる場面においては、1秒あたりの打鍵の頻度が0である時刻が多いことが読みとれる。また、打鍵の頻度が0である時間は、注視点が画面内にある場合が多いことが読みとれる。

このようなパターンが現れた原因について考えてみると、悩んでいる場面においては、被験者は、次に成すべき作業が分からなくなり、画面をじっと眺めている状態にあると予想される。

このようなパターンを検索するために、“1秒あたりの打鍵の頻度が0である状態が n 秒続き、かつ、注視点が画面内にある状態が n 秒以上続く”という場面を表す次のような検索文を作成した。

{key: $x = 0, t \geq n$ } and
{eye: $x = 1, t \geq n$ }

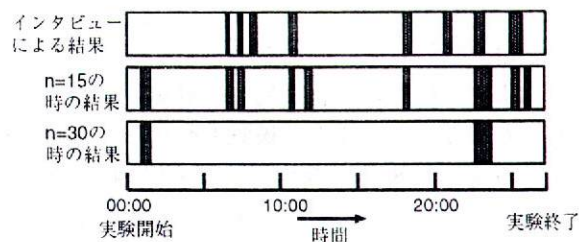


図6: 結果の比較

インタビューにより特定した被験者の悩んでいる時刻と、作成した検索文を用いて検索した結果特定した時刻が、どの程度一致するかを調べた。検索文における n の値を、15、および、30に設定した場合の検索結果との比較結果を図6に示す。 $n = 30$ の場合は、検索された場面数が2場面と少なく、インタビューによって特定された場面と1場面しか一致していない。 $n = 15$ の場合は場面数も多く(9場面)、分析者により挙げられた8場面のうち、6場面が一致している。 n の値を14以下にした場合には、悩んでいる場面以外の場面を数多く検索してしまったため、採用しなかった。

キーストロークの頻度と、注視点が画面内/外にあるという単純な行動履歴データからだけでも、“悩んでいる”のような被験者の内面的な行動でさえもある程度特定できることが確認できた。

検索文により特定できなかった悩んでいる場面は、図5における場面3および場面6であった。場面3では、ごく少ない打鍵があった。また、場面6は、時おり注視点が画面外に移動していた。

一方、検索文により誤って特定された悩んでいる場面は、 $n = 15$ の時は3場面あった。その3場面においては、被験者は考え込んでいたが、悩んでいるはいなかった。

5.4 考察

実験結果は、3.1節で述べた“開発者の特定の行動、および、特定の状態変化は、行動履歴データの特定の値の変化に対応している”という仮定に矛盾しない。本システムを用いて方法1および2による開発行動の分析が可能であることを示す結果が得られたと言える。また、“悩んでいる”のような開発者の内面的な行動についても、行動履歴データだけからある程度特定できることが分かった。

6 まとめ

本稿では、ソフトウェア開発行動を支援するための2通りの方法を提案し、分析を支援するための

システムについて述べた。実験による評価の結果、分析者は、行動履歴データを表示したグラフを見ることにより、被験者の特徴的な行動をいくつか特定することができた。また、被験者へのインタビューにより特定した8箇所の“悩んでいる”場面のうち、6箇所をシステムにより検索することができた。本システムを用いることにより、ソフトウェア開発行動の分析に要する時間を短縮できることを示唆する結果が得られたといえる。

ただし、1回の試行しか行えていないために、システムを用いた2通りの分析方法が常に有効であるとは言えない。異なる条件の下での実験を重ねて、システムを有効性についてさらに検討する必要がある。

また、複数の状態の遷移を表すようなデータに対して検索を行う場合には、状態遷移の順序を指定して検索できる必要があり、より複雑な条件での検索が行えるように、検索方法を改良することが考えられる。

参考文献

- [1] 旭敏之: “ユーザの操作履歴をもとに、使いにくさを出荷前に改善”, 日経エレクトロニクス, No. 609, pp. 111-120 (1994).
- [2] Ginger2 Home Page: URL “<http://hawk.aist-nara.ac.jp/ginger2/>”
- [3] 飯尾和彦, 新井義夫, 古山恒夫: “モジュール間の視点の移動に着目したプログラム理解過程の実験と分析”, 人工知能学会研究会資料, SIG-KBS-9402, pp. 9-16 (1994).
- [4] 小林淳, 下條善史, 松本健一, 鳥居宏次: “ソフトウェアの開発/利用行動を記録した映像に対する自動索引付け法の提案”, 情報処理学会研究報告, 96-SE-107, pp. 33-40 (1996).
- [5] Koenemann J., Robertson S. P.: “Expert problem solving strategies for program comprehension,” Proceedings of CHI’88 Conference on Human Factors in Computing Systems ACM, pp. 125-130 (1988).
- [6] 三輪和久, 杉江昇: “学習の初期段階における計算機プログラミングの動的過程” 人工知能学会誌, Vol. 7, No. 1, pp. 138-148 (1992)
- [7] 門田暁人, 高田義広, 鳥居宏次: “視線追跡装置を用いたデバッグプロセスの観察実験”, 電子情報通信学会技術報告, SS96-5, pp. 1-8 (1996).
- [8] 森孝宏, 西田知博, 齊藤明紀, 都倉信樹: “大量のGUI操作履歴を分析するための走査・再生ツール”, 情報処理学会研究報告, 96-HI-69, pp. 1-8 (1996).
- [9] 本杉匡史: “プログラム保守におけるデバッグ行動の分析”, 修士論文, 奈良先端科学技術大学院大学, NAIST-IS-MT9451113 (1995).
- [10] 中島田義敬, 森崎修司, 門田暁人, 松本健一, 鳥居宏次: “ソフトウェア開発行動の記録映像を検索するシステムの試作”, 情報処理学会第53回全国大会, 1D-3, pp. 1-195-1-196 (1996).
- [11] 高田義広, 鳥居宏次: “プログラム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, Vol. J77-D-I, No. 9, pp. 646-655 (1994).
- [12] 柳正純, 門田暁人, 高田義広, 鳥居宏次: “バグ混入時のプログラミング行動の特徴を検出するツールの試作”, 電子情報通信学会技術報告, SS94-36, pp. 9-16 (1994).
- [13] 吉川厚: “プログラム理解過程を調べた認知実験報告(1)”, 人工知能学会研究会資料, SIG-IES-9202, pp. 53-61 (1992).