

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

小林 淳†‡ 下條 善史† 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 8916-5

‡ 東電ソフトウェア株式会社 設計開発センター
〒 105 東京都港区新橋 6-19-15 東京美術倶楽部ビル

E-mail: {atusi-k, zen-s, matumoto, torii}@is.aist-nara.ac.jp

ソフトウェアの品質や生産性の向上を目的として、ソフトウェアの開発/利用行動をビデオ映像として記録し、行動の分析を行う研究が行われている。本稿では、被写体である人間（被験者）による打鍵やマウスの移動、ファイル更新履歴等の補助情報（ビデオ映像と並行収集された情報）を利用し、ソフトウェアの開発/利用行動に特有な事象や被験者の状態変化が発生した時刻を索引としてビデオ映像に自動的に付加する方法を提案する。更に、提案する方法の有効性を検証するために行った思考実験の結果について報告する。提案する方法によって付加された索引を利用すれば、被験者が特定の行動を行った時点やその前後の映像の検索が容易になり、ビデオ映像に基づく行動分析に要する時間が大幅に短縮されることが期待される。

An Indexing Method for Video Recordings of Software Development and Software Usage

Atushi Kobayashi†‡ Zenshi Shimojo† Ken-ichi Matsumoto† Koji Torii†

† Graduate School of Information Science, Nara Institute of Science and Technology
‡ Toden Software Incorporation

† 8916-5 Takayama, Ikoma, Nara 630-01, Japan

‡ 6-19-15 Shinbashi, Minato, Tokyo 105, Japan

An indexing method is proposed for video recordings of activities that occur during the development and usage of software. This method is based on events that are characteristic in the development and usage of software. In other words, features of human-computer interaction are employed, such as data concerning keystroke, mouse events, and file savings. Since analysis of video recording is currently done manually, this method is expected to enable a drastic reduction in time spent on analysis.

1 はじめに

ソフトウェアの品質や生産性の向上を目的として、ソフトウェアの開発/利用行動の分析が盛んに行なわれている [10] [14]. 行動分析の代表的な方法としては、被験者の操作状況を記録するモニタリングや、操作時に考えていることを被験者に発話してもらい、その内容を解析するプロトコル分析などがある [1]. いずれの方法においても、被験者の動作や周囲の状況を記録することができるビデオが利用されることが多い [4].

ビデオ映像には被験者の動作や周囲の状況が記録されているが、その一部を自動的に抽出するために利用できる情報(索引)は、時刻情報以外含まれていないのが一般的である. 従って、ビデオ映像を用いた分析においては、分析者が映像を一覧して、分析に必要と思われる映像の記録された時刻のリストを作成する. しかし、数時間から数日にわたって記録された大量のビデオ映像を一覧することは容易ではない.

索引付けを自動的に行う方法としては、画像処理技術を用いる方法や補助情報を用いる方法が知られている [3][7][12] [15]. しかし、これらの方法は適用分野を限定しており、ソフトウェアの開発/利用行動の分析にそのまま用いることはできない. 例えば、画像処理技術を用いる方法では、隣接する映像フレーム間での画像特徴の変化を検出し、その時刻を場面やカットの切り替わり時刻として索引付けする. ソフトウェアの開発/利用行動を記録した映像では、場面やカットの切り替わりは少なく、十分な索引付けは期待できない. また、補助情報を用いる方法では、ビデオ映像の記録開始/中断/終了時点や被写体を変更した時点で、カメラの状態、撮影地点の地理的位置、撮影方位、撮影時刻、記録テープの位置等の情報を同時に記録し、索引として利用する. ソフトウェアの開発/利用行動を対象とした場合、映像記録の中断や被写体の変更はあまり行われないので、十分な索引付けは期待できない.

ソフトウェアの開発/利用行動の大きな特徴は、被写体となる人間(被験者)が計算機と情報を交換しながら作業を進める点にある. そして、被験者と計算機間で交換される情報を操作履歴として収集する方法やシステムが、既にいくつか提案されている点である [2][5][6] [11] (図1参照). 操作履歴は画像処理技術を用いることなく自動収集可能で、しかも、ビデオ映像と並行して秒単位で収集可能である. 即ち、ソフトウェアの開発/利用行動を記録した映像を対象とした場合、操作履歴は、従来の索引付け法で用いられていた情報よりも収集が容易で、かつ、詳細な索引付けに利用できる

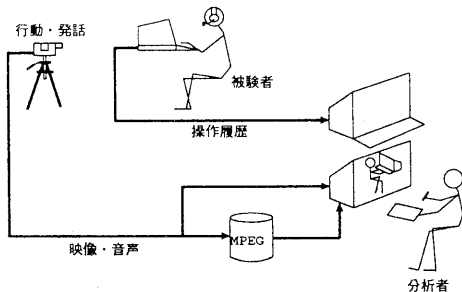


図 1: ソフトウェア開発/利用行動観測実験

情報といえる.

本稿では、被写体である人間(被験者)による打鍵やマウスの移動、ファイル更新履歴といった操作履歴をビデオ映像の補助情報として利用し、被験者が特定の行動を行った時点やその前後の映像を検索するために必要な索引をビデオ映像に自動的に付加する方法を提案する. 更に、提案する方法の妥当性を検証するために行った思考実験の結果について報告する. 提案する方法では、ソフトウェア開発/利用行動を状態遷移モデルとして定義している. これにより、ソフトウェアの開発/利用行動に特有な事象や被験者の状態変化が発生した時刻を補助情報から特定し、索引としてビデオ映像に付加することが可能となっている.

以下 2 章では、従来の索引付け法の概要を紹介し、ソフトウェアの開発/利用行動を記録した映像にそれらを適用した場合の問題点について述べる. 3 章では、提案する方法による索引付け、及び、映像検索の手順とそこで用いられる情報(データ)の構造について説明する. 4 章では、提案する方法の妥当性を評価するために行った思考実験の概要と結果について述べる.

2 従来の索引付け法

2.1 画像処理技術を用いる方法—ニュース放送映像への索引付け

画像処理技術を用いて、カットの変わり目の自動検出による索引付け法が提案されている [7][12] [15]. この方法では、フレームを周期的に抽出し、隣接するフレームが互いに異なる画像特徴を持っている時点を探し出し、その時点のカットの変わり目とし、カット単位に分割する. その際の索引は各カットの始まりから数秒間を繰り返し表示する micon を利用する. micon はそのカットを代表する情報であり、一覧して表示することにより、映像の索引としている [7].

画像処理技術を用いて索引付けを行う研究としては、

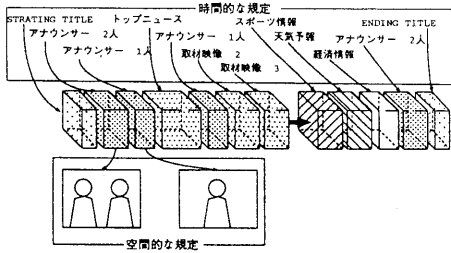


図 2: ニュース映像の特徴

対象となる映像をニュース映像に限定したものがある [12] [15].

ニュース映像にはその構成が時間的にも空間的にもある程度規定されているという特徴がある。時間的な規定とは、図 2 に示すように、数種類のショット (画面の切り替わりのないひとまとまりの映像) の時系列でニュース番組が構成されていることを指す。一方、空間的な規定とは、アナウンサーのショットにおける画面上の位置などが一定であることを指す。

こうした制約の下で、画像処理技術を用いることにより、一連の映像を単なるカット割でない、論理的に意味のある時点で分割することが可能となる。また、時間的な規定と空間的な規定以外にも、ショット中に表示される字幕を索引語に用いる研究も行われている。

2.2 補助情報を用いる方法—資料映像に対する索引付け

補助情報を用いる方法は、人文科学の研究者が研究対象となる資料を映像データとして記録し、分析するためのシステムでよく用いられている [3].

補助情報は、映像の撮影時にカメラの録画ボタンが押された時点で自動的に付加される。これは、人文科学の研究の場合、被写体が歴史的資料 (例えば、寺院や古文書) である場合が多く、録画ボタンが押される毎に、新しい対象の撮影が始まることが期待されるためである。補助情報のうち、地理的位置 (東経, 北緯), 撮影方位, 年月日, 時刻は GPS (Global Positioning System) により記録される。

2.3 従来法を適用した場合の問題点

従来の索引付け法をソフトウェア開発/利用行動の映像に適用した場合の問題点は次の通りである。

- ソフトウェア開発/利用行動を記録は、主に固定カメラによって被験者の机上での作業や着座の様子を撮影したものであるため、大きな動作が少なく、場面やカットの切り替わりも少ない。従っ

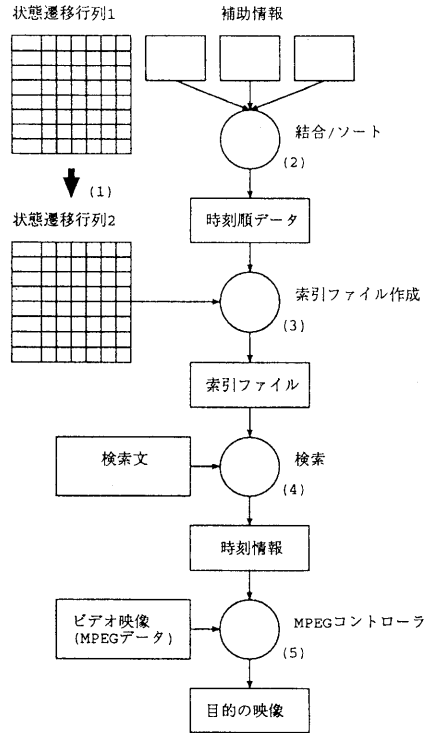


図 3: 索引付けと検索の手順

て、画像処理技術を用いる方法は適していない。

- ソフトウェア開発/利用行動を記録する際には、被験者への侵襲性の問題より、撮影は実験開始と同時に始まり、実験終了時に撮影も終了する。従って、録画ボタンが押された時点で補助情報が記録される従来の方式は適していない。

3 提案する索引付け法

3.1 索引付けと検索の手順

提案する方法における索引付けと検索の手順を図 3 に示す。図 3 中の矩形はデータを、円はデータの変換モジュールを、矢印はデータの流れをそれぞれ表す。提案する方法では、ビデオ映像、ビデオ映像と並行収集された情報 (補助情報), 及びソフトウェア開発/利用行動を定義した状態遷移モデル (状態遷移行列) を入力とし、検索文で指定された条件に合致する時刻や期間の映像を出力として得る。以下では処理の概要を述べる。(図 3 中の番号は処理番号 (1)~(5) に対応する.)

- (1) ソフトウェア開発/利用行動を状態遷移モデル(状態遷移行列1)として定義する。更に、状態遷移行列1中の各遷移条件を補助情報に対する条件に変換した状態遷移モデル(状態遷移行列2)を作成する。
- (2) 補助情報の格納されたファイルを結合し、時刻順にソートして時刻順データを作成する。
- (3) 時刻順データに状態遷移行列2をあてはめて索引ファイルを作成する。
- (4) 検索文をもとに索引ファイル内を検索し、目的とする映像が記録された時刻や期間のリストを時刻情報として作成する。
- (5) 時刻情報をMPEGコントローラに送り、目的の映像を出力として得る。

以下では、索引付けと検索に用いる情報(データ)とその生成過程について詳細に説明する。

3.2 ソフトウェア開発/利用行動の状態遷移モデル

図4は、ソフトウェア開発/利用行動のうち、開発・保守作業を状態遷移モデルとして表現したものである。モデルの記法は、オブジェクト指向のモデル化方法論であるOMTに従っている[9]。ここではソフトウェア開発/利用行動におけるオブジェクトとして開発者/利用者たる人間と、人間が操作する計算機の二つを定めた。この両者が相互作用しながら開発/利用を遂行していく様子をシナリオとし、これに基づいて人間の状態と、状態の遷移条件である事象を定めた。

図4のモデルを表形式で表現したものが「状態遷移行列1」である。例として、図4中の状態「テスト」に関わる部分の状態遷移行列1を表1に示す。この行列の要素 a_{ij} は、状態 i から状態 j への遷移に必要な事象を表す。 a_{ij} が“-”とは、状態 i から状態 j への遷移が定義されていないことを表す。

「状態遷移行列2」は、状態遷移行列1の各要素を「時刻順データ」に関する条件(遷移に必要な時刻順データの項目)に置き換えたものである。表1に対応する状態遷移行列2を表2に示す。表2中の“*”は時刻順データが存在すればその内容に関わらず遷移することを、“ ϕ ”は時刻順データの終端で遷移することを、それぞれ表す。

3.3 索引ファイルの作成手順

図3の補助情報としては、被験者による打鍵データ、計算機上でのプロセスの起動履歴、計算機による表示メッセージ等を考える。これらのデータには、形式はそれぞれ異なるが、収集時刻が必ず含まれている。

補助情報は結合、ソートされて時刻順データとなる。

時刻順データには補助情報として収集された全てのデータに対して、収集時刻、データ種別、値が時刻順に並べ換えられて含まれている。時刻順データの例を図5に示す。

この例では、“key”は打鍵データを、“proc”はOS上のプロセスのステータスを、“err”は対話画面の標準エラー出力に送られたデータを、“mouse”はマウスに対する操作を、“out”は標準出力に送られたデータを、それぞれ表している。

時刻順データは先頭のデータから順に状態遷移行列2の遷移条件(事象)と比較され、その結果として状態の遷移履歴が作成される。例えば、図5の時刻順データを表2に適用した場合、次のような遷移履歴が得られる。(但し、キー操作(key a.out testdata<RET>)によって、10:11:39に状態「実行オブジェクト入力」となっているものとする。)

10:11:44 “out Execution complete”, 即ち、事象「終了メッセージが出力される」の発生によって状態「テスト結果参照」に遷移する

10:11:48 “key cat result<RET>”, 即ち、事象「テスト結果表示」の発生によって状態「判断」に遷移する

このようにして作成された遷移履歴を格納する索引ファイルの例を図6に示す。

3.4 検索インタフェース

検索するための入力は、状態と事象からなる系列である。出力は、入力された系列の開始時刻と終了時刻の組合せである。複数回、入力された系列が存在する場合には存在する数だけ、入力された系列の開始時刻と終了時刻の組が出力される。

入力となる状態と事象の系列は、状態や事象のみでなく、状態から事象へ、事象から状態へ、更には、状態から隣接しない状態への遷移を表現可能とすることを目標とする。入力の記述は正規表現[8]を基とする、

状態 [(属性) [(属性)...]]

事象 [(属性) [(属性)...]] [[ガード]]

から構成される元 α は連接 $\{XY|X \in \alpha, Y \in \alpha\}$ と論理和 $\{X \cup Y|X \in \alpha \text{ or } Y \in \alpha\}$ と反復 $X^* = X$ の0回以上の繰り返しの組合せで表現される。共通属性は時刻である。

例えば、“編集～コンパイル～編集という行動系列が起こったコンパイル前の編集行動の終りから5分前の行動”という系列は次のように記述する。

編集, (ファイル更新(ファイル名))

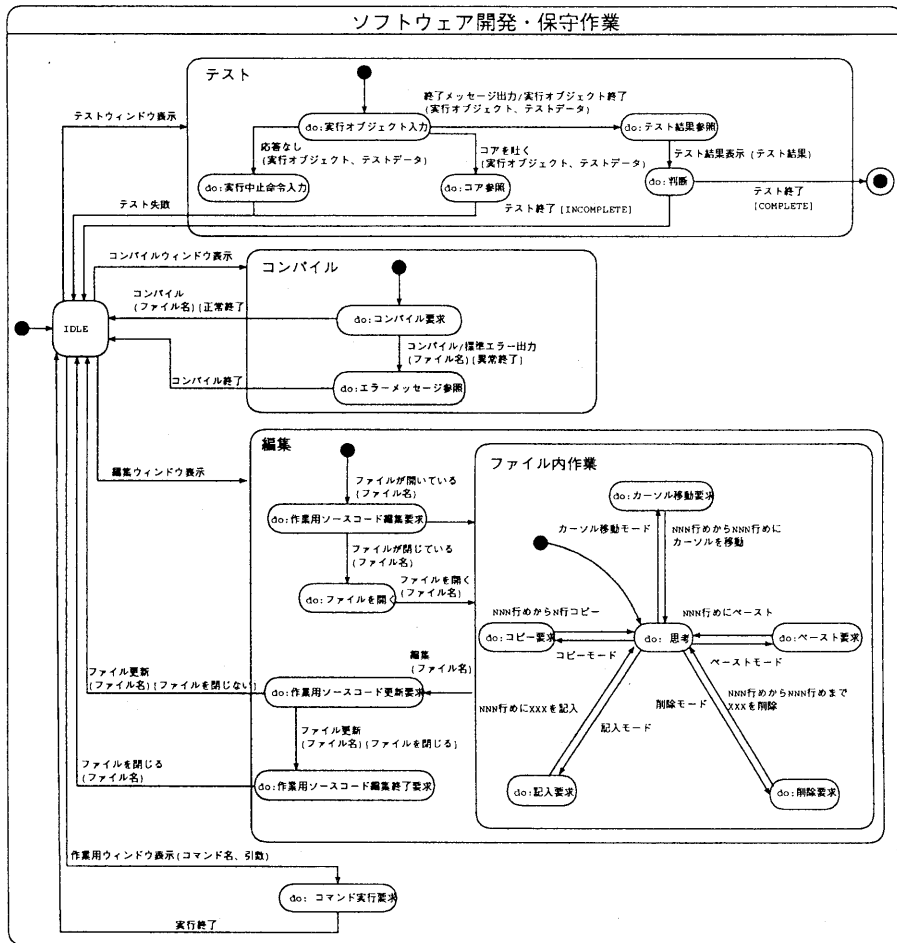


図 4: 被験者の状態遷移図

[ファイルを閉じない] + ファイルを閉じる
(ファイル名) (5分), IDLE, コンパイル,
IDLE, 編集

この記述では、ある編集作業が行われ、5分後にファイル更新、もしくはファイルが閉じられ、コンパイルが行われ、編集作業が行われたことが表されている。コンパイル状態から IDLE 状態への事象は 2 種類存在するがどちらの事象を経て IDLE 状態へ達したかは、問われていないので、その間の事象を記述していない。明示的に記述するならば、

コンパイル, (コンパイル (ファイル名)
[正常終了] + コンパイル終了), IDLE

もしくは,
コンパイル, -, IDLE

である。

正規化表現の拡張として、状態や事象、事象中の属性を特定しない場合、'-' と記述する。'X-Y' は状態 (事象)'X' から状態 (事象)'Y' へ何らかの事象 (状態) を通過して達したことを表す。例えば、図 4 中の状態「ファイル内作業」内の事象「NNN 行に XXX を記入」を持ちいて検索する場合、'NNN' に数式を、XXX には文字を記入する。数式や文字を特定したくない場合には '-' を用いる。なお、数式記入箇所では範囲での指定を可能とする。例えば、23 行めから、5、6 行削除ならば、

23 行めから 23+(5..6) 行めまで - を削除とする。

入力された系列は、索引ファイルとパターン・マッチング処理が行われ、マッチすると系列の開始時刻と終了時刻の組を出力する。

表 1: 「テスト」の状態遷移行列 1

	実行オブジェクト入力	実行中止命令入力	コア参照	テスト結果参照	判断	IDLE	終了
実行オブジェクト入力	—	応答なし	コアを吐く	終了メッセージ出力	—	—	—
実行中止命令入力	—	—	—	—	—	テスト終了 (INCOMPLETE)	—
コア参照	—	—	—	—	—	テスト終了 (INCOMPLETE)	—
テスト結果参照	—	—	—	—	テスト結果表示	—	—
判断	—	—	—	—	—	テスト終了 (INCOMPLETE)	テスト終了 (COMPLETE)
終了	—	—	—	—	—	—	—

表 2: 「テスト」の状態遷移行列 2

	実行オブジェクト入力	実行中止命令入力	コア参照	テスト結果参照	判断	IDLE	終了
実行オブジェクト入力	—	プロセス “a.out” が 3 分以上実行状態にある	ファイル “core” が存在する	終了メッセージが出力される	—	—	—
実行中止命令入力	—	—	—	—	—	*	—
コア参照	—	—	—	—	—	*	—
テスト結果参照	—	—	—	—	プロセス “cat result” が正常終了	—	—
判断	—	—	—	—	—	*	φ
終了	—	—	—	—	—	—	—

4 思考実験

提案した方法によってソフトウェア開発/利用行動を観察する分析者が、その実験目的に応じた映像の検索を簡易に行なえることを評価するために、思考実験とその結果の分析を行なった。

4.1 実験の概要

思考実験の手順は以下の通りである。

- (1) ソフトウェア開発/利用行動について映像を利用した観察を行なっている研究者に対して、各々がどのような場面を観察しているか、または観察したいかをインタビューし、検索対象となる場面を 17 場面収集した。具体的にはソフトウェア開発、ソフトウェア保守、ソフトウェア利用の各作業に注目している研究者 3 名から場面リストを得た。
- (2) 収集したリストの各場面に対して、本方法による検索が可能かどうかを机上にてシミュレートした。
- (3) 検索が不可能な場合、それを解決する方法を考察した。

3 名の研究者は、それぞれ以下のような目的でソフトウェア開発/利用行動の観察を行っている。

- ソフトウェア開発時において、プログラミング行動中のバグの混入時の特徴的な行動の有無を調べる。

- ソフトウェア保守作業において、混入されたバグの除去作業の手順を明らかにする。
- ソフトウェア利用行動における、操作の効率を低下させる原因や誤った行動を引き起こす原因を特定する。

4.2 収集された場面リスト

3 名の研究者が挙げた、検索対象となる場面は以下の通りである。

(1) ソフトウェア開発

1. 特定回数内の修正
2. バグ混入時の特徴的な行動パターン [13]
 - a. 1~20 行程度がコピーされ、別の位置にペーストした後に、その数箇所に変更を加える
 - b. 1 行を入力途中で、別の位置の変更後、入力が再開される
 - c. 同じ行の同じ箇所が何度も変更される
 - d. 1 行入力後、その行の 1 箇所を変更した後に、大きく移動後、別の作業を行う
 - e. 1 行の入力に 1 分以上かかる
 - f. 元々、誤りの入っていた行が別の場所にコピーされる

(2) ソフトウェア保守

1. コンパイル直前の編集作業

```

10:11:30 key   cc source.c<RET>

        proc  Is+ 0:00.60 tcsh
              S  0:53.65 vi source.c
              R+ 0:00.03 cc source.c

10:11:38 err   Compile Complete

        mouse test active

10:11:39 key   a.out testdata<RET>

10:11:40 proc  Is+ 0:03.37 tcsh
              S  0:58.21 vi source.c
              R+ 0:00.15 a.out testdata2

10:11:44 out   Execution complete

10:11:48 Key   cat result<RET>

        out   Test Data Caliculation
              #0001  0.0000001
              #0002  0.0000002

```

図 5: 時刻順データの例

```

10:11:38 上位状態[テスト]
10:11:39 状態[実行オブジェクト入力]
        key   a.out testdata<RET>
10:11:44 事象[終了メッセージ出力]
        out   Execution Complete
10:11:48 状態[テスト結果参照]
        key   cat result<RET>
10:11:48 事象[テスト結果表示]
        out   Test Data Caliculation
              #0001  0.0000001
              #0002  0.0000002
10:11:48 状態[判断]

```

図 6: 索引ファイルの例

2. 保守作業終了直前に修正した箇所をそれ以前に修正した
3. 編集開始からコンパイルまでの間に同じ行の同じ箇所が何度も変更される
4. 編集→コンパイル→実行の順以外で作業を行なう
5. "オッ"と言った
6. 1度修正した箇所を元に戻す
7. 編集中で一定時間以上のキー操作なし
8. 離席直前の行動

(3) ソフトウェア利用

1. ユーザの誤操作
2. ヘルプ画面参照直前作業
3. ユーザが一定時間操作をしていない
4. 必要以上の繰り返し操作

4.3 検索記述

上記 17 の検索対象行動系列を検索言語で記述した。

- 1.1 カーソル移動要求, 一行めから " 関数先頭行~関数終了行 " 行めにカーソルを移動, -
- 1.2.a コピー要求, 一行めから (1...20) 行コピー, 思考, カーソル移動要求, 一行めから (-+-) 行めにカーソルを移動, 思考, ペースト要求, 一行めに, ペースト, 思考, (記入+削除)
- 1.2.d 一行めに-を記入, 思考, (削除要求+ペースト要求), 思考,

カーソル移動要求, 一行めから (-+20) 行めにカーソルを移動

- 1.2.e 記入要求 (1分)
- 2.1 -, 作業用ソースコード更新要求, IDLE, コンパイル
- 2.2 編集, IDLE, テスト, IDLE, コンパイル
- 2.7 思考 (1分)
- 3.2 作業用ウィンドウ表示 (HELP)
- 3.3 (IDLE(1分)+思考(1分))
- 3.4 (作業用ウィンドウ表示, コマンド実行, 実行終了, IDLE)³
- 4.4 考察

4.3 章であげたものは, 本方法で提案する状態遷移図上での場面指定が可能であった。したがって本方法による索引付けが可能であり, 有効な観察環境が提供できると考えられる。しかし, 残りのものについては状態遷移図によって場面の指定が不可能であり, したがって本方法による索引付けが不可能なものと分類される。

思考実験の結果, 本索引付け法について以下のような問題点が見出された。

- (1) 「ある行を編集した後で, 再び同じ行を編集した。」というような検索が行なえない。これは, 同一の行でも, その行の上に別の行が作成されれば, 行番号が変わるからである。
- (2) ファイルの編集に偏ったモデルになっており, エディタ以外のソフトウェアを利用する場合のモ

デルが不十分である。

- (3)「編集, コンパイル, テストという正規の順番以外の順で作業が行なわれた。」というような検索が行なえない。検索文の文法で「以外の」というような記述を許していないためである。

これらのうち問題点(1)については, 行の挿入や削除が起こっても, 行の認識ができるように, 行識別子を属性として持たせることにより解決できる。また, 他の問題点についても, 状態遷移モデルや検索言語の文法の拡張で対応可能と思われる。そこで, 今回の問題点から以下のような拡張が考えられる。

- 編集, コンパイル, テストというような単純な開発方法だけでなく, デバッガなどの開発ツールの使用を前提としたモデルへの拡張が望まれる。
- 検索言語の記述能力が正規表現と一部の拡張だけでは不足することがあるので, より高い記述能力を持った言語へと拡張することが望まれる。

また本索引付け法は, 映像に対する索引というだけでなく, 例えば発話プロトコル法における発話データへの索引や, 人間の生理データなどのようにデータの構造がよく理解されていない時系列データに対する索引付けにも利用可能と思われる。

5 おわりに

本稿では, 被写体である人間(被験者)による打鍵やマウスの移動, ファイル更新履歴といった操作履歴をビデオ映像の補助情報として利用し, 被験者が特定の行動を行った時点やその前後の映像を検索するために必要な索引をビデオ映像に自動的に付加する方法を提案した。提案する方法では, ソフトウェア開発/利用行動を状態遷移モデルとして定義している。これにより, ソフトウェアの開発/利用行動に特有な事象や被験者の状態変化が発生した時刻を補助情報から特定し, 索引としてビデオ映像に付加することが可能となっている。

本稿では, 更に, 提案した方法の妥当性を検証するためにに行った思考実験の結果について報告した。思考実験においては, ソフトウェア開発/利用行動をビデオ映像を用いて分析した経験を持つ3名を対象としてインタビューを行い, 映像の分析において検索するであろう典型的な場面を列挙してもらった。列挙された17場面のうち10場面については, 提案する方法により映像に付加した索引に基づいて検索可能であることが分かった。

今後の課題としては, ソフトウェア開発/利用行動を表す状態遷移モデルをより汎用的なものとすると共

に, 提案する方法に基づくビデオ映像検索システムを試作し, 方法の有効性を実験的に検証することが考えられる。

参考文献

- [1] 旭敏之, 岡田英彦, 井関治, 松田良一: “ユーザの操作履歴をもとに, 使いにくさを出荷前に改善”, NIKKEI ELECTRONICS, No.609, pp.111-120, 1994.
- [2] 海老名毅, 伊藤昭: “Xアプリケーションにおける可読性のある操作履歴の取得について”, 電子情報通信学会技術研究報告, HC94-87, pp.1-7, 1995.
- [3] 星野聡: “映像データの収集利用方式の開発”, 情報処理学会研究報告, 93-CH-20, pp.21-28, 1993.
- [4] 海保博之, 原田悦子: “プロトコル分析入門 発話データから何を読むか”, 新曜社, 1993.
- [5] Shinji Kusumoto, Ken-ichi Matsumoto, Tohru Kikuno and Koji Torii: “On a measurement environment for controlling software development activities”, IEICE Transactions on Communications Electronics Information and Systems, E74, 5, pp.1051-1054, 1991.
- [6] 森川治: “対話型システムにおけるタイミング情報を含むキー操作列の解釈法について”, 電子情報通信学会論文誌, J70-D, 11, pp.2198-2203, 1987.
- [7] 長坂晃朗, 田中謙: “カラービデオ映像における自動索引付け法と物体探索”, 情報処理学会論文誌, 33, 4, pp.543-550, 1992.
- [8] J. ホップクロフト, J. ウルマン共著, 野崎昭弘, 高橋正子, 町田元, 山崎秀記共訳: “オートマTON言語理論 計算論 I”, サイエンス社, 1994.
- [9] J. ランボー, M. プラハ, W. プレメラニ, F. エディ, W. ローレンセン著, 羽生田栄一監訳: “オブジェクト思考方法論 OMT”, トップラン, 1992.
- [10] Jeffrey Rubin: “HANDBOOK OF USABILITY TESTING: How to Plan, and Conduct Effective Tests”, JOHN WILEY & SONS, INC., 1994.
- [11] 高田, 鳥居: “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, J77-D-I, 9, pp.646-655 (1994).
- [12] 山本浩司, 浅野正一郎: “ニュース映像データベースの索引づけ手法の一提案”, 情報処理学会 第51回全国大会, 6S-10, 1995.
- [13] 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研究報告, SS94-36, pp.9-16, 1994.
- [14] 吉川厚: “プログラム理解過程を調べた認知実験報告”, 人工知能学会研究会資料, SIG-IES-9202-7, pp.53-61, 1992.
- [15] H.J.Zhang, Y.Gong, S.W.Smoliar, S.Y.Tan: “Automatic Parsing of News Video”, Proc.IEEE Int'l Conf.Multimedia Computing and Systems, pp.45-54, 1994.