

多段差分生成によるユーザー操作記録の自動比較

来住伸子

日本アイ・ビー・エム(株) 東京基礎研究所

ユーザー操作の記録をとることによって、作業効率や使い易さの評価を行うことが試みられているが、記録データの解析、特にデータの意味付けに時間がかかることが多い。そこで、最近のウィンドウ・システムに標準となりつつあるユーザー操作の記録および再生機能を利用して、ユーザー操作記録をとり、その操作記録を自動的に比較するツールを試作してみた。今回の発表では、ツールで使用した多段差分生成法を主に説明し、その実験結果について考察する。

Automatic Analysis of User Operation Records with Multi-Step Matching

Nobuko Kishi

Tokyo Research Laboratory, IBM Japan Ltd.
5-19 Sanbancho, Chiyoda-ku, Tokyo 102, Japan

Recorded data of users' operations are not used effectively during usability evaluation processes because of the high cost of analysis. We developed a set of tools, named SimUI, for evaluating graphical user interface designs by automating analysis of user operation records. These tools record mouse and keyboard operations by users and compare them with model operations to detect potential usability problems. This paper describes one of SimUI's main techniques, multi-step matching of recorded data, and discusses the result of a preliminary experiment.

1 はじめに

ユーザーインターフェイス評価方法の目的として、使い易さの目安を提供することがあげられるが、ソフトウェア開発中に使用可能な評価方法ということを目指した場合、時間や労力をできるだけ少なくすませ、具体的な問題点を指摘するというような条件を満たすことも要求される。

そこで、この2条件を満たすユーザーインターフェイス評価方法を目指して、ユーザー操作記録の比較を自動的に行うことを試みることにした。設計者が行った作業の操作記録と、実際のユーザーによる操作記録の違いを検出することによって、設計者が意図した使い方とユーザーが解釈した使い方との違いを具体的に指摘できるようになる。また、自動的に行うことによって、ユーザーインターフェイス評価をより正確にかつ効率的に行うことができるようになる。

この操作記録の自動比較を行うツール、SimUI (SIMulated User Interaction analyzer) を現在試作中である。図1は、SimUIの比較結果表示の一方法を示している。右上の3個の円と、その下の3個の半円が比較結果をアニメーションで表示する。左下にあるのが、ユーザー操作の対象となるアプリケーションのウィンドウである。比較結果と、ユーザーの操作記録を同時に再生している。

円の方は、操作記録が一致しているかどうかを表示しており、一致なら緑、余分な操作なら黄色、不一致なら赤になる。半円の方は、一致した操作間の時間の比較結果を表示しており、針が中心にあれば同じ速度、左にあれば基準となる操作記録より遅く、右にあればより早く操作を入力していることになる。

2 背景

ユーザー操作の記録の取り方としては、たとえば、

1. 音声録音 — ユーザーの発言内容
2. デジタル — キーボードの打鍵記録
3. ビデオ — ユーザーの振る舞い

などがあげられる。これらの記録方法には、それぞれ一長一短がある。一番早くから行われたのは、

1のユーザーの話したことを録音しておく方法で、録音しておいたものを文章にし、プロトコル分析などを使用して解析する。その次はユーザーの入力を計算機で記録するもので、打鍵レベルモデルなどがその解析方法の一例である。最近使われることが多いのはビデオでユーザーの様子を撮影および録音しておくというもので、この場合は必ずしも解析方法が確立しているわけではなく、人間の観察者の判断の補助にするという使い方が主であることが多い。これは、グラフィカルユーザーインターフェイスへの入力記録を解釈するというのはかなり困難で、それよりも、ユーザーの表情や画面の状態を目でみて判断した方が手軽だということに原因がある。

このように、いろいろな記録方法があるのは操作記録技術の進展のためと言えるが、もう一つの考え方として、評価の対象となる、ユーザーの情報処理の仕組みのレベルが異なるためとも言える。たとえば、プロトコル分析は、認知的なレベルの情報処理に関心がある時に使用され、発言記録は、ある情報処理と次の情報処理の間のユーザーの知識状態を表現する系列として解釈されることが多い。打鍵レベルモデルは、認知レベルでは知覚されない段階の情報処理に関心がある時に使用され、操作時間によって、思考が必要な操作とそうでない操作を分類することが、ユーザーの振る舞いの分析の出発点となっている。そして、ビデオ記録は、ユーザーの振る舞いの認知レベルも操作レベルも総合的に観察しようという試みであると考えられる。(実際には、観察者の主観によって表情や動作の観察を行なうことが多いので、分析手法としては、前二者ほどには確立されていないが。)

したがって、これらの解析方法に共通する仮定として、次の2点があげられる。

- 人間の情報処理の仕組みには、高次レベルと低次レベルのいくつかの階層構造がある。
- ユーザーが計算機を理想的な使い方で行っているときには、状況に合ったレベルの情報処理を行っている。

そこで、操作記録の解析にはどのようなレベルの処理をユーザーが行っているかということを設定することが重要なことがわかる。たとえば、レベルの同定には、プロトコル分析では発言内容、打

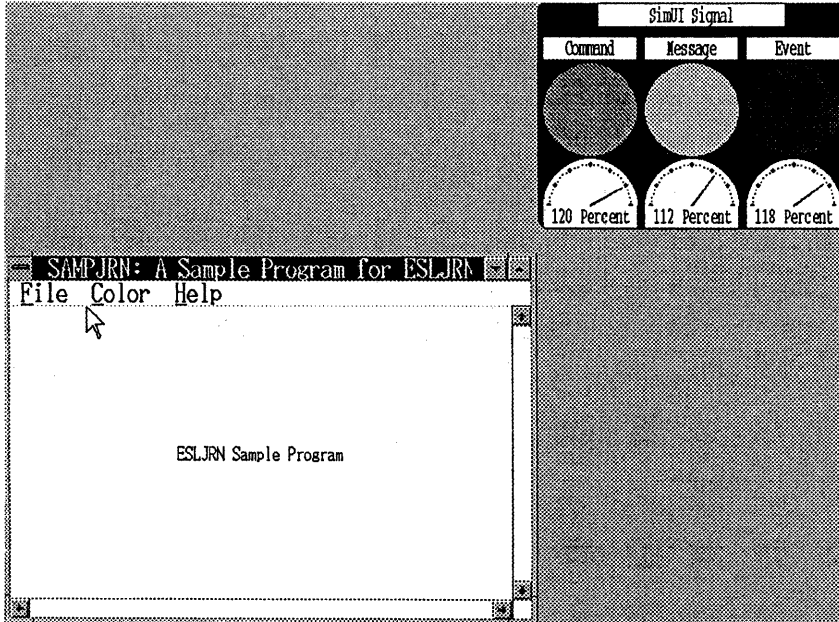


図 1: 多段差分生成による比較結果表示アニメーション

鍵レベルモデルでは時間が使用されていると考えられる。今回、グラフィカルユーザーインターフェイスに対する操作記録の解析を試みるにあたって、ウィンドウシステムの状態を利用して、レベルの同定を行うことにした。つまり、ユーザーがマウスを動かすとかキーを押すとかの入力を行ったときに、ウィンドウシステムの側でどんな種類のメッセージが起きるかによってユーザー操作の重要度を推定することにした。

このようにウィンドウシステムの状態を利用した操作記録の解析を行うために多段差分生成法という手法を考えてみた。この多段差分生成法によって、ユーザーの操作記録から次のような事象を自動的に抽出することができた。

- 基準となる使い方と異なる使い方をしている。
- 基準となる使い方と一致しているが、異なるスピードで起きている。

このような事象を自動的に抽出することが、ユーザーインターフェイス評価をすべて自動化するわけではないが、評価の生産性を高めるツールとしてかなり有効な手法であると考えられる。

3 SimUI の構成

SimUI の主要なツールには次のようなものがある。

- 記録ツール
- アプリケーション修正ツール
- 再生/データ収集ツール
- 記録比較ツール (多段差分生成ツール)
- 比較結果表示ツール

これらのツールは図 2 に示した順に使用する。まず、記録ツールによって、アプリケーションを使うときのユーザーの操作を記録する (Step 1)。このときは再生に最小限必要なデータのみを記録する。次に、アプリケーション修正ツールによって、アプリケーションをバイナリすなわち実行可能モジュールの形のまままで変更を加える。

再生ツールは、変更を加えたアプリケーションに、記録しておいたユーザー入力を与えることによって、アプリケーションの振り舞いを再生する (Step 2)。このとき、データ収集ツールが、ユーザー操作だけでなく、アプリケーションの実行状態を知るための情報も記録しておく。

記録比較ツールは、このようにして集めた操作記録を二つ、多段差分生成法によって比較するこ

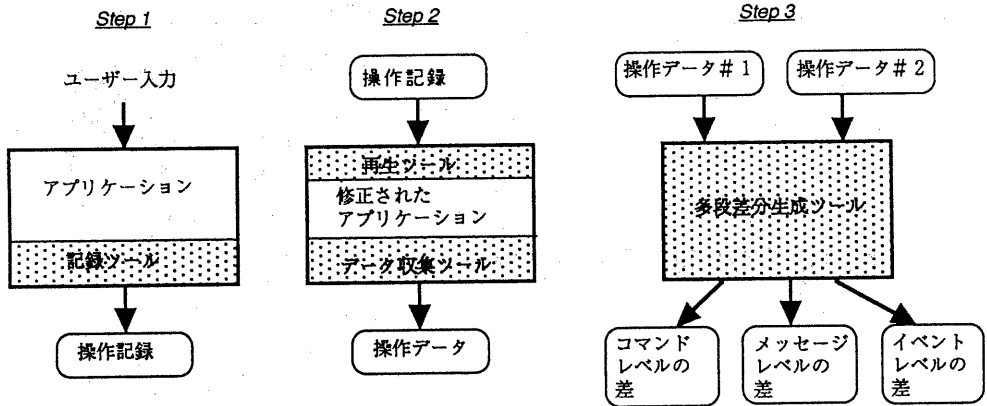


図 2: SimUI の主要なツール

とにより、ユーザーインターフェイス評価に意味のある差分を生成する (Step 3)。

図 1 は比較結果表示ツールの画面で、アプリケーションの振る舞いを再生しながら、生成した差分をアニメーションで表示する。

SimUI の技術的特徴としては、次のような 2 点がある。

- 再生時のデータ収集
- 多段差分生成

このうち、前者については既に [2] で発表しているので、今回は後者について主に説明する。

4 操作記録の収集

OS/2 Presentation Manager や X Window のようなマルチプロセスのもとでのウィンドウシステムでは、ウィンドウシステムとアプリケーション間のデータの受け渡しやサービスの依頼はメッセージによって行われる [3]。ウィンドウシステムとアプリケーションの間の処理の同期は、いろいろな種類のプロセス間通信を使ってとられる。特にユーザー入力の場合、送信側がメッセージをメッセージ列 (queue) においた後、受信側が適当な時期にメッセージ列から取り出すという形で同期がとられることが多い。

OS/2 Presentation Manager では、システムメッセージ列 (system queue) と呼ばれる特別なメッセージ列が一つあり、タイマー、キーボード、マウスといったデバイスからの入力、メッセージ

としてここにおかれる。システムメッセージ列におかれたメッセージは、OS/2 Presentation Manager の振り分け部によって、各ウィンドウに振り分けられる。

たとえばキーボードからの入力は、押したり離したりした時刻、物理キーコード、論理キーコード、シフト状態という情報を含むメッセージとしてシステムメッセージ列に置かれる。振り分け部は、ウィンドウの状態に基づいて、キーボード入力を受け取るべきウィンドウをきめ、そのウィンドウ処理するプロセスが入力を要求したときに、そのウィンドウを所有するアプリケーションメッセージ列経由で渡す。

SimUI のデータ収集ツールは、システムメッセージ列におかれたメッセージと、記録の対象としたいアプリケーションに属するアプリケーションメッセージ列に置かれたメッセージの両方を記録する (図 3)。ユーザーの入力した操作の記録には、システムメッセージ列におかれたメッセージの記録だけでも十分であるが、後での解析の際に、アプリケーションの実行状態の情報も必要となってくるので、アプリケーションメッセージ列のメッセージも記録している。

Listing 1 は、このようにしてデータ収集ツールが記録したデータの一部である。各行が一つの記録の単位で、先頭が小文字ではじまる語は予約語、大文字で始まる語や、HWNDnnnn:nnnn で表された数字は識別名、// より後ろは注釈という構文になっている。予約語 map で始まる記録は、新しいオブジェクト、たとえば、プッシュボタンのよう

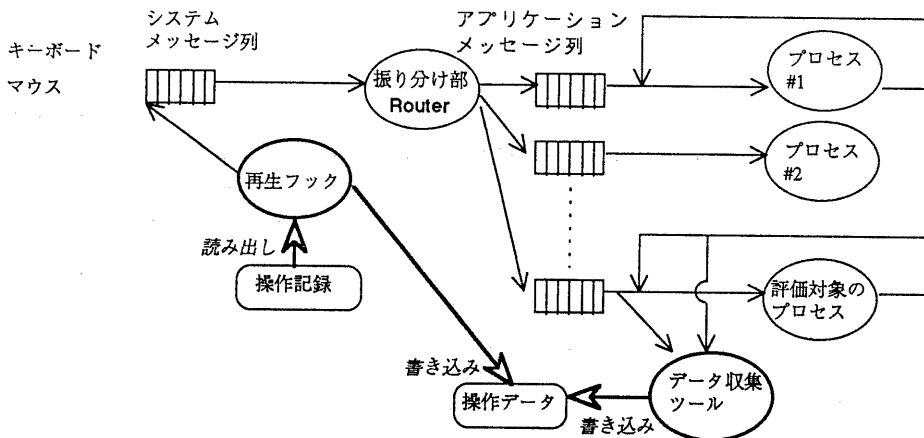


図 3: ユーザー操作の再生とデータ収集

なユーザーインターフェイス要素やウィンドウ本体がつくられたときに生成される。予約語 `send` で始まる記録は、システムメッセージ列からメッセージが取り除かれたときに生成される。予約語 `receive` で始まる記録は、アプリケーションが、メッセージを何らかの方法で受け取ったときに生成される。先頭が大文字で始まる語は識別名で、特に WM で始まる名前はメッセージのタイプ名を表現している。時間の記録は差分生成の比較の対象としないが、操作記録間の相対的な速度の計算をするために、注釈部分に入れている。time の後の数字は、ユーザーが実際に入力した時刻、ptime の後の数字は、SimUI のデータ収集ツールが入力記録を再生した時刻をミリ秒単位で表現している。前者の数字は、システムクロックの制約から約 30 ミリ秒、後者の数字は、ファイル I/O などのデータ収集にともなう負荷のために約 300 ミリ秒の誤差を含んでいる。

5 操作記録の自動比較

5.1 差分生成

Listing 1 からわかるように、SimUI の収集するデータは一つのメッセージが一行になる形式になっている。これは、ファイルの行単位の比較によって、メッセージ列の比較をし、差分生成をするためである。ファイルの行単位の比較は、既存のファイル差分生成ツール、UNIX 系の環境では `diff` というような名前で行われるツールが利

用できる。SimUI が使用したのは SuperCompare という名前のもので、元々 10 年程前に MVS 用に製品化されたツールを OS/2 上に移植したものである [1]。SuperCompare の特長としては、

- 動的計画法ではなくハッシングを使用しているので、よく似たデータ同志ならば、かなり大きなデータを高速に処理できる。
- 連続する空白文字や // や * で始まる注釈を比較の対象から外すという指定が可能である。
- 指定した語を含む行を比較の対象から外すフィルタという機能がある。

などが挙げられる。

5.2 多段差分生成

ファイル差分生成ツールによって、実際に差分生成してみると、観察者が感じる違いとかなる異なる部分が、差分として生成されてしまう。これは、ユーザーの入力記録が次のような性質を持っているためである。

- 改行キーや空白キーのように何度も繰り返して起こるイベントがある。
- マウスポインタの XY 座標が異なると、観察者には殆ど同じ位置にみえても、異なる値のパラメータのイベントになってしまう。

そこで、人間の観察者にとって自然な差分を生成するために、フィルタを何種類か用意して、観察

Listing 1: SimUI のデータ収集ツールが生成するデータ例

```
map HWND5000:3A04 to WindowHWND3 id 32776 in WindowHWND2 // ClassName "EaselClientClass"
map HWND4FB0:3928 to MenuHWND1 in WindowHWND2 // System Menu in WinCreateStdWindow()
map HWND4FA0:37A4 to MenuHWND2 in WindowHWND2 // Title Bar in WinCreateStdWindow()
map HWND4FF0:3974 to MenuHWND3 in WindowHWND2 // Horizontal Scroll Bar in WinCreateStdWindow()
.....
send WM_MOUSEMOVE from SysQ x 220 y 198 // ptime 251781 // mp1 12976348 mp2 0 time 3229343
receive WM_BUTTON1DOWN by WindowHWND6 x 46 y 13 // ptime 252125 //
send WM_BUTTON1UP from SysQ x 220 y 198 // ptime 252125 // mp1 12976348 mp2 1022 time 3229531
receive WM_MOUSEMOVE by WindowHWND6 x 45 y 12 // ptime 252125 //
send WM_MOUSEMOVE from SysQ x 220 y 198 // ptime 252375 // mp1 12976348 mp2 0 time 3229625
receive WM_BUTTON1UP by WindowHWND6 x 45 y 12 // ptime 252375 //
receive WM_QUERYFOCUSCHAIN by WindowHWND3 id 5 // ptime 252448 //
```

者に意味のあるイベントから順に、差分生成を行う、多段差分生成を試みることにした。多段差分生成を大まかに説明すると、初めはユーザーに影響の大きいイベントについてのみ差分生成を行い、徐々に差分生成の対象となるイベントを増やしながらか差分生成を繰り返していく。また、差分生成で一致したイベント同士に共通の通算番号をふるることによって、後の差分生成で他のイベントと一致することを防ぐ。

具体的には、次のような処理をフィルタの個数分だけ繰り返すことにより行う。ここで、お手本となる操作記録をファイル A、ユーザーの操作記録をファイル B とする。

1. 一回目の差分生成のときは、ファイル A、B をそのままファイル A',B' として使用する。二回目以降の差分生成のときは、前回の差分生成によって生成されているファイル A', ファイル B' を使用する。
2. A' と B' とフィルタをファイル差分生成プログラムに与える。ファイル差分生成プログラムは A' と B' の各行を読む度にフィルタを参照し、フィルタに記述してある語を含む場合は、差分生成の対象としない。
3. 差分生成プログラムの出力を解析し、一致したとみなされた行には、通算番号をふる。こ

の通算番号を元のファイル A, B の先頭に付けたものを、次の差分生成のときのファイル A',B' として使用する。

5.3 フィルタの種類

SimUI では、今のところ 3 個のフィルタを次の順に使用している。

1. コマンドレベル・フィルタ
ユーザーが選択したコマンドの識別番号をアプリケーションに知らせるためのメッセージを通すフィルタ。このフィルタによって、ユーザーが異なる手段(たとえば、プルダウン・メニューと押しボタン)でコマンドを選択していても、コマンドの識別番号さえ同じならば、同じメッセージがアプリケーションに渡される。
2. メッセージレベル・フィルタ
コマンドレベル・フィルタが通すメッセージに加えて、各ユーザーインターフェイス要素の状態をアプリケーションに知らせるためのメッセージを通すフィルタ。このフィルタによって、ユーザーが異なる入力方法(たとえば、マウスとキーボード)を使用していても、同じユーザーインターフェイス要素を選んでいれば、同じメッセージがアプリケーションに渡される。

3. イベントレベル・フィルタ

上記の二つのフィルタが通すメッセージに加えて、システムメッセージ列から渡されるメッセージを通すフィルタ。このフィルタによって、ユーザー入力操作そのものが比較の対象になる。

5.4 記録比較ツール

記録比較ツールは、図 4 のような 3 つのウィンドウを横に並べて表示し、多段差分生成を対話的に行うツールである。左端のウィンドウは左列に比較結果、中央列と右列に操作記録の行番号を表示している。比較結果は、左から順に、コマンドレベル、メッセージレベル、イベントレベルの比較結果を、次のような文字で表示したものである。

- M 手本とユーザーの記録データが一致した。
- C 手本とユーザーの記録データが一致していない。
- I ユーザーの記録データに余分な部分がある。
- D ユーザーの記録データに足りない部分がある。

(空白) 比較の対象にできなかった。

中央のウィンドウは、見本とすべき記録データ、右端のウィンドウは、ユーザーの記録データを表示している。実際には、一致しているところには緑、異なっているところには赤、余分な部分には黄色というように色を付けてある。

5.5 比較結果表示ツール

図 1 は比較結果表示ツールの画面で、3 個の円は、記録比較ツールで使用した色を各レベルごとに、ユーザーの操作記録の再生に合わせて表示している。半円の方は、一致したイベント間の時間間隔が、基準となる操作記録に対してどれ位の比率になるかを % で示している。中心が 100 % に対応している。

6 多段差分生成の使用実験

SimUI の多段差分生成という方法が、どの程度ユーザーインターフェイス評価に有効かを調べる

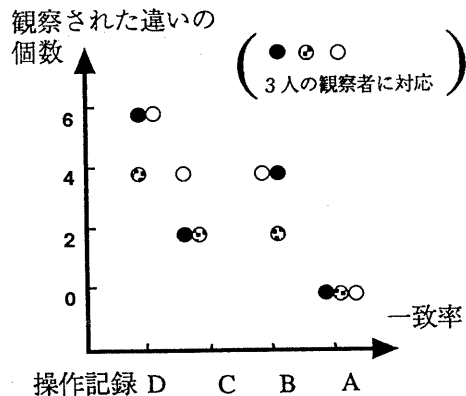


図 5: 被験者による観察と一致率の関係

表 1: 実験に使用した操作記録の一致率 (%)

操作記録	D	C	B	A
コマンドレベル	44	68	100	100
メッセージレベル	40	42	61	100
イベントレベル	0	0	0	100

ために、簡単な実験を行ってみた。まず、次のように 4 つの操作記録を用意した。

- A ある作業を間違いなく終えた時の操作
- B 作業は正しいが異なる人がゆっくりと行った操作
- C コマンドを何回か間違えた時の操作
- D 作業は正しいがヘルプなど余分なコマンドを含む操作

これらの操作記録の、操作記録 A を基準としたときの一致率は、表 1 のようになる。一致率とは、各レベルの対象となるイベントのなかで、基準となる記録と一致したイベントが占める割合である。

3 人の被験者に、まず操作記録 A の再生を見本として見せ、次に A から D の操作記録を取り混ぜて見せ、できるだけ多く、見本と異なる点を指摘してもらった。

結果は、図 5 のようになる。この図から次のようなことが観察できる。

1. 同じ操作記録から別々に再生したものを異なるといった人は一人もいなかった (操作記録 A)。

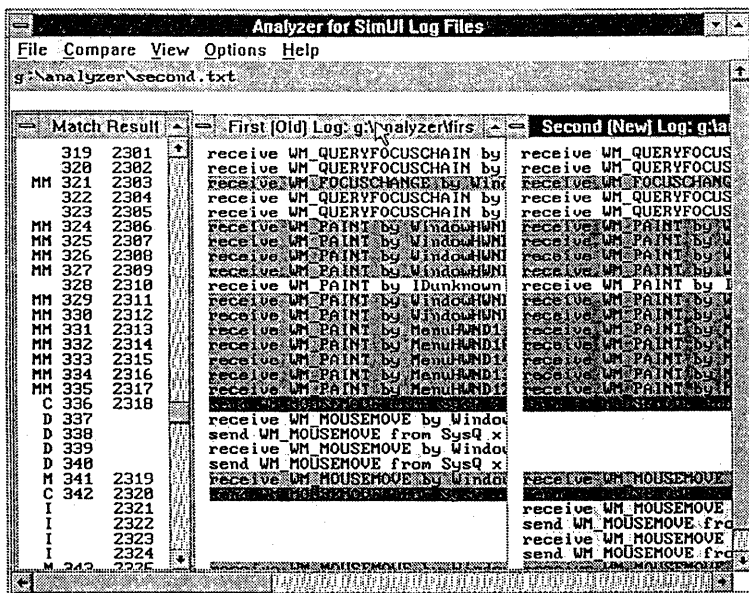


図 4: SimUI の記録比較ツール

2. コマンドレベルの不一致でも気が付かない人がいた (操作記録 C と D)。
3. メッセージレベルの一致率が低くても、人間がみると一致しているようにみえることがある (操作記録 B と C)。

つまり、観察者が見落としていた問題点を見つけるという効果は認められたが、観察者が問題と思う個所の個数と、多段差分生成による一致率は必ずしも比例するというわけではないことがわかった。

7 まとめと今後の方向

現在、SimUI は OS/2 1.3 上での試作を終え、ようやく安定して使用できるようになってきた段階である。この段階でのまとめとしては、次のようなことが上げられる。

- 多段差分生成は、ユーザーインターフェイス評価において観察する人の見落としを防ぐという手法として有効なことが確認できた。観察者が気づくことはすべて検出できるが、観察者が問題ではないと思うような点も差分として生成することがある。この点については別のタイプのフィルタを追加することによって改善を図る予定である。

- SimUI における多段差分生成の実装の仕方には技術的な課題がまだある。たとえば、12,3 秒程度の操作記録を比較するのに、i486DX 33Mhz で、5 分程かかる。これは、フィルタによる前処理と差分生成とを別にするることによって短縮する予定である。

このように、実用化のための作業は多く残されているが、当面は、SimUI ツールが実際のユーザーインターフェイス評価にどの程度役立つかを実験を通じて確認していく予定である。多段差分生成による一致率がユーザーインターフェイス評価の基準の一つになるレベルまで、将来試みてみたい。

参考文献

- [1] IBM Corp., Source Compare/Audit Utility Program Description/Operation Manual, SH20-2361. IBM, 1979.
- [2] 来住伸子, ユーザー操作の再生機能を利用した UI 評価の試み. 情報処理学会第 33 回プログラミング・シンポジウム報告集, pp135-143, 1992.
- [3] Microsoft Corp., Microsoft OS/2 Programmer's Reference. Microsoft Press, 1989.