

利用ログに対するウィンドウ遷移の解析による 操作学習システム生成

岩田 一^{†1} 白銀 純子^{†2} 深澤 良彰^{†3}

ソフトウェアの操作方法の学習においては、業務でよく利用される操作をユーザに提示し、学習させることが効果的である。しかし、ある機能を利用するための操作の流れが開発者の意図したものと、多くのユーザが利用しているものとで一致しているとは限らない。そこで本研究では、ソフトウェアの効果的な操作学習のために、利用ログから頻度の高い操作を解析し、学習支援システムを構築する手法を提案する。

Generating Operation Learning System Based on Window Switching Analysis Using Log Files

HAJIME IWATA^{†1}, JUNKO SHIROGANE^{†2}
and YOSHIKI FUKAZAWA^{†3}

For learning software operation, it is effective for end-users to learn frequently performed operation methods. However, users' actual operations are not necessarily ones expected by developers. In our research, we propose a method for generating a tutorial system by analyzing operation log files that end-users operate software and by extracting window sequences.

1. はじめに

多種多様な業務のコンピュータ化により、ソフトウェアにおいてユーザが利用する必要の

ある機能が増えてきている。そのため、ソフトウェアを導入しても、必要な作業を行うために学習しなければならない操作が多く、操作方法の習得が困難な現状がある。この解決策の1つとして、ユーザがソフトウェアの操作の流れを学習することを支援する手法が、これまでに提案・実現されている。

ソフトウェアの操作の流れを、ユーザがより効率良く学習を行うことができるようにするためには、実際のソフトウェアのGUI(Graphical User Interface)を用いて、操作の手順を実利用により近い形で示すことが重要である。そのためには、実際のソフトウェアのGUIを利用して、操作方法の提示を行うチュートリアルシステムが適している。チュートリアルシステムは、1つ1つの詳細な手順を順に示すことにより、ユーザの目的を達成するために必要な操作を、初心者であっても容易に理解することを目指すシステムである¹⁾。ユーザは、ソフトウェアの機能の利用に必要な一連の操作内容を、実ソフトウェアに沿って確認し、学習を行う。チュートリアルシステムでは、常にユーザの実利用に近い形で表示されるため、学習支援システムで示される内容と実ソフトウェアとの間の乖離が少ない。またチュートリアルシステムでは、ソフトウェアの操作方法をある程度の単位ごとに提示し、ユーザの操作によって提示内容を切り替えるため、ユーザが自分の進捗で学習を行うことができる。そこで、チュートリアルシステムを用いた学習支援に注目する。

多くの機能を持つソフトウェアの操作方法の学習においては、業務でよく利用される操作をユーザに提示し、学習させることが効果的である。そのためには、ユーザがどのような操作の流れでソフトウェアを操作するかを分析する必要がある。また、ソフトウェアの動作に沿ってチュートリアルシステムを動作させるためには、対象となるソフトウェアとチュートリアルシステムとの連携が必要である。しかし、ある機能を利用するための操作の流れが開発者の意図したものと、多くのユーザが利用しているものとで一致しているとは限らない。

そこで本手法では、チュートリアルシステムの作成およびソフトウェアへの付加の労力を軽減することを目的とする。チュートリアルシステムでは、業務で頻繁に利用される操作をユーザに提示することが重要である。そのために、ユーザのソフトウェアに対する操作のログを解析することにより、チュートリアルシステムを自動生成し、ソフトウェアに付加する。

2. 本手法の特徴

本手法では、ユーザの操作ログを解析することにより、チュートリアルシステムを生成し、ソフトウェアに付加を行う。

GUIを有するソフトウェアは、通常様々な機能を利用するための、操作の起点となるウイ

^{†1} 神奈川工科大学
Kanagawa Institute of Technology
^{†2} 東京女子大学
Tokyo Woman's Christian University
^{†3} 早稲田大学
Waseda University

ンドウが存在する．このウィンドウを本手法ではメインウィンドウと呼ぶ．本手法において想定しているソフトウェアは，このメインウィンドウがソフトウェア起動時に最初に表示されており，メインウィンドウから各機能を実行するものである．

本手法で想定するソフトウェアにおいて，ユーザが操作を行い目的とする機能を実行する過程は，以下ようになる．

- (1) メインウィンドウに配置されているボタンやメニューなどをユーザが選択することにより，それぞれの処理を進めるためのメインウィンドウとは別のウィンドウが表示される．
- (2) (1) で表示されたウィンドウに配置されているウィジェットを操作することにより，さらに別のウィンドウに遷移していく．
- (3) 必要な処理が終了すると，ウィンドウの遷移も終了し，それまでに表示されていたメインウィンドウ以外のウィンドウは閉じられて，メインウィンドウに戻る．

本手法では，この完結する操作手順の単位を判断するために，膨大な操作ログの中からウィンドウの遷移に着目する．メインウィンドウから始まり，ウィンドウの遷移をたどり，メインウィンドウに戻ってくるまでの一手順を，1つの操作手順の単位であると判断する．本手法では，この操作手順の単位を完結フローと呼ぶ．ただし，完結フローは“操作の開始から，ある1つの目的を達成して操作が終了するまでの操作手順”を意味するため，操作の流れが途中から始まっていたり，途中で終わっていたりするようなものは，完結フローではない．

完結フローの例として，図1に住所録ソフトウェアの例を挙げる．図1において，左側に表示されているウィンドウが，住所録ソフトウェアで実現される「登録」「閲覧」などの機能を実行するための起点となる，メインウィンドウである．メインウィンドウの「登録」ボタンをクリックすることで，住所情報を登録するためのウィンドウが新規に表示される．登録する人物の住所情報を入力し，「登録」ボタンをクリックすることで登録の作業を完了し，登録のためのウィンドウが閉じられることによって，ユーザは再びメインウィンドウで操作を行うことになる．このメインウィンドウを起点として，いくつかのウィンドウが表示され，閉じられてメインウィンドウに戻るまでの一連の遷移が完結フローである．

完結フローの中から，抽出した操作手順のまとまりを比較し，完結フローのうち同じ操作手順となっているものの利用頻度を求める．これは頻繁に利用されている操作手順を，ユーザにとって重要なものであると考えからである．利用頻度の高い完結フローを開発者に提示し，学習する操作として適切なものを選択してもらう．また，完結フローはある特定の組

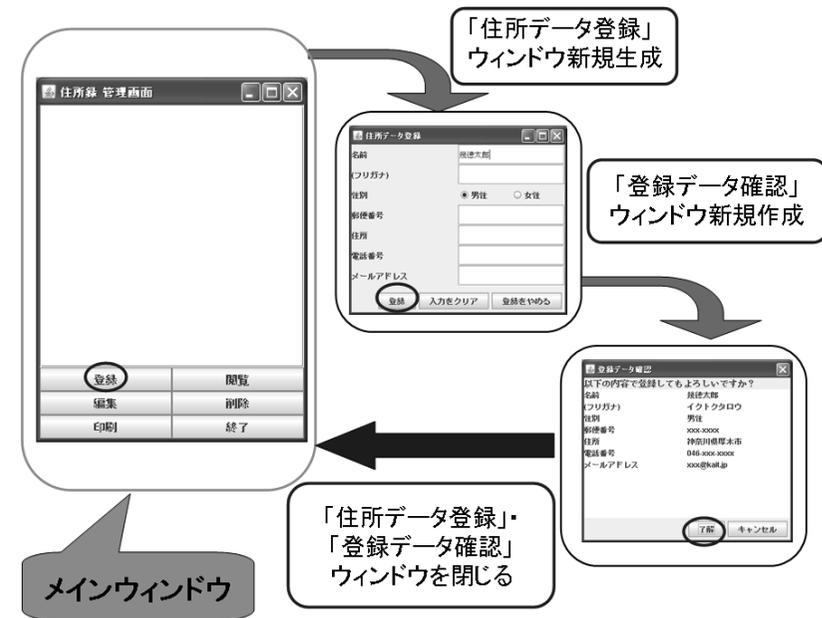


図1 「住所情報の登録」の完結フロー
Fig. 1 An unit flow of Address Registration

み合わせでよく使われるものが考えられる．例えば，住所録ソフトウェアにおいて住所を登録した後，あて名の印刷を行う，といった一連の手順が挙げられる．そこで，開発者に選択された完結フローから，組み合わせられて利用されるものを抽出する．その完結フローの組み合わせの操作手順をチュートリアルとして，ユーザに提示する．

3. チュートリアルシステムの例

図2に，住所録ソフトウェアに対する本手法で想定するチュートリアルシステムの例を示す．図2の右側が，チュートリアルシステムのウィンドウであり，左側がチュートリアルの対象となる，住所録ソフトウェアのウィンドウである．チュートリアルとしてユーザが学習したい一連の操作の流れを，コンボボックスの中から選択し，チュートリアルシステムのウィンドウの中にある，“チュートリアル”ボタンをクリックすることで動作する．

図2における例では，住所録ソフトウェアに対して，チュートリアル“「登録 閲覧 編

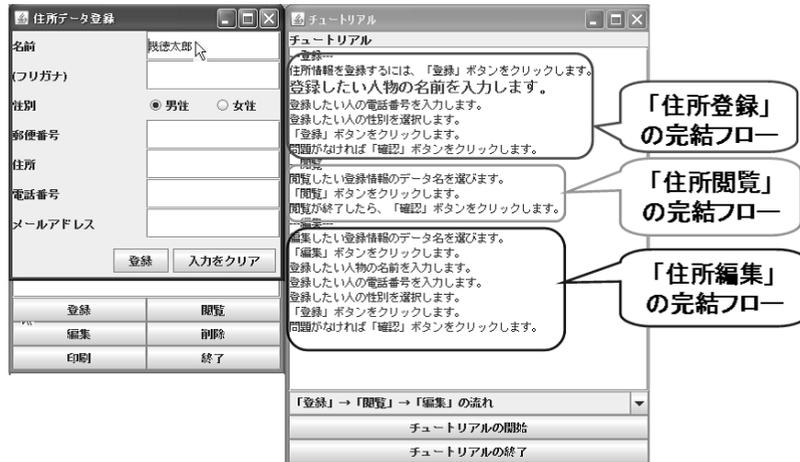


図2 住所録における「登録 閲覧 編集」のチュートリアル
Fig.2 An example of tutorial system

集」」を実行することで、住所情報の登録・閲覧および登録された情報の編集までの一連の操作の流れを表示している。チュートリアルシステムのウィンドウには、現在提示している作業の説明として、「登録したい人物の名前を入力します。」が強調表示されている。また、ここでは登録する人物の名前の例として、「幾徳太郎」がチュートリアルシステムにより自動的に入力されている。そして、その入力作業が行われる名前のテキストフィールドに対してマウスポインタが表示されている。このようにして、チュートリアルシステムはその時々で必要となる操作内容をユーザに対して示す。これらの手順は、チュートリアルシステムのウィンドウ上において箇条書きで示されており、チュートリアルシステムにより順番に実行され、その過程がユーザに対して提示される。また、「住所情報の登録」についてのチュートリアルした後、同様にして「住所情報の閲覧」、「住所情報の編集」を続けて提示することによって、3つの機能の一連の流れがユーザに提示される。

4. システムの構成

本手法におけるシステムの概要図を図3に示す。本手法を用いて、チュートリアルシステムを生成して付加する手順は以下の通りである。

- (1) ログイングツールの付加
- (2) 完結フローの解析
- (3) 操作手順の解析
- (4) 完結フローの選択・提示用データ・提示文の記述
- (5) チュートリアルシステムの生成
- (6) チュートリアルシステムの付加

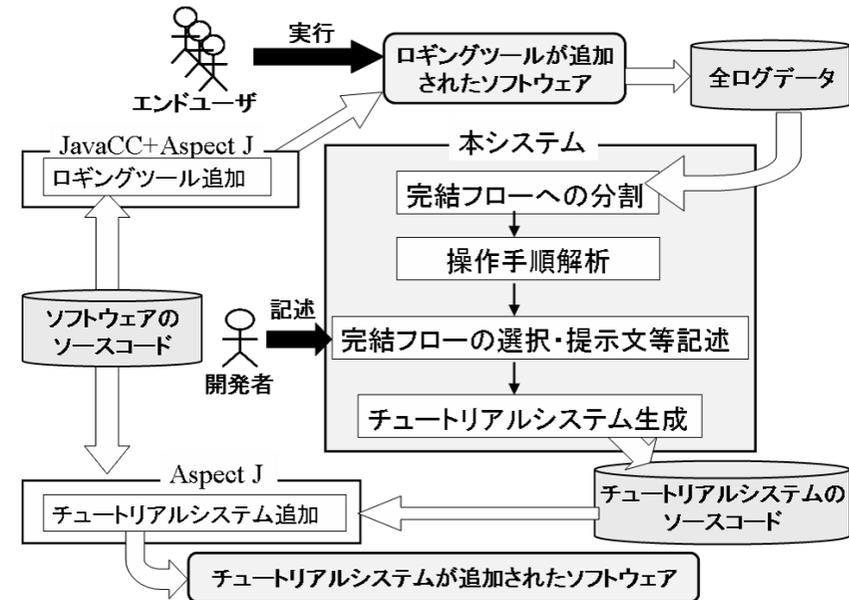


図3 システム構成図
Fig.3 System architecture

4.1 ログイングツールの付加

本手法におけるログイングツールとは、ソフトウェアに存在するウィジェットに対して、ユーザが操作を行うことによって発生するアクションをログとして保存するものである。アクションとは、ユーザがウィジェットに対して行った操作のことである。例えば、ボタンに対するアクションとしてはクリック、テキストフィールドに対しては文字入力などである。

チュートリアルシステムを自動生成するために、ロギングツールにより取得するログは以下の通りである。

- エンドユーザによるアクションが発生した時間
- エンドユーザによるアクションが発生した際にアクティブになっているウィンドウ
- エンドユーザによるアクションの対象となるウィジェットの種類
- エンドユーザによるアクションの対象となるウィジェットの変数名
- エンドユーザによるアクションの内容
- エンドユーザによるアクションの対象となったウィジェットのラベル

ソフトウェアに対してロギングツールを付加するためには、ソフトウェアのウィジェットの変数名をシステムが把握することが必要である。そのために、JavaCC(Java Compiler Compiler)²⁾によって生成されたJavaソースコード解析ツールを利用する。JavaCCとは、Javaで記述されたソースコードの字句解析、構文解析を行うためのツールである。

本手法では、対象となるソフトウェアのソースコードを解析し、ソースコード内に記述されているウィンドウを生成するクラスの名前と、そのウィンドウの中に存在するウィジェットの種類と変数名を抽出する。この抽出した結果をもとにして、ソフトウェアのウィジェットに対して上記の情報をログとして出力するロギングツールを生成する。ロギングツールは、AspectJ³⁾のソースコードとして生成する。

操作ログの取得機能は、ソフトウェアの機能とは別のものであり、また、アクションが発生する全てのウィジェットに対して付加しなければならないものである。ウィジェットの定義等の処理はソースコードの広範囲にわたって記述されているため、この操作ログの取得機能をアスペクトとして記述することが適切である。したがって、ソースコードの解析によって得られるウィジェットの変数名をもとに、ロギングツールのアスペクトを生成する。このアスペクトをソフトウェアにウィーブすることにより、ユーザが行った操作ログを取得することができる。

4.2 操作ログからの完結フローの抽出

4.1節で述べたロギングツールを付加したソフトウェアをユーザが利用することによって操作ログを得る。この操作ログをもとに、エンドユーザによるアクションが発生した時間を利用して時系列に沿って操作ログを解析する。

完結フローを抽出するために、まずユーザアクションが発生した時にアクティブになっているウィンドウの名前を取り出す。これにより、ウィンドウの名前が変わるときにウィンドウの遷移が発生していることがわかるため、ウィンドウごとの遷移を単位とした解析を行う

ことができる。

GUIでの操作では、例えばテキストフィールドに対する文章の修正や、選択項目に対する選び直しなど、1つのウィジェットに対して複数回の操作が行われることも多い。操作ログには、そのような細かな操作も全て記録されるため、膨大なログが生成されてしまう。しかしながら、チュートリアルとしては、1文字1文字の入力や選択項目の選択しなおしなどの、1つのウィジェットに対する複数回の操作の記録は必要ではなく、1つのウィジェットに対しては1回分の操作の記録があればよい。

そこで本手法では、まず記録されているウィジェットに関するログをウィンドウごとにとめる。これは、操作の流れをウィンドウ単位で判断することによって、1つ1つのウィジェットに対して発生する細かな操作ではなく、操作全体の流れを得るためである。またログには、ユーザが操作した各ウィジェットの変数名と、ボタンクリックやキーボード入力などの操作の内容が記録されている。同一ウィンドウ上のそれぞれのウィジェットに対して発生した操作ログについては、操作の内容と操作が発生したことのみを記録として残し、行われた操作の回数については残さないこととする。

操作ログを完結フローとしてまとめる際には、操作の開始は「メインウィンドウ上でボタン等のアクションが発生した場合」とし、操作の終了は「他の全てのウィンドウを閉じ、起点となるメインウィンドウに戻ってきた場合」とする。また、操作中に新しいウィンドウを開き続ける限りは「操作は継続中」と判断する。このように判断することにより、ソフトウェア起動時のメインウィンドウから、ユーザが操作を行いそれぞれのウィンドウの遷移が発生し、再びメインウィンドウに戻ってくるまでの操作の流れを、1つの完結フローとして考えることができる。

4.3 操作手順の解析

4.2節で抽出した完結フローを組み合わせ、チュートリアルで利用する操作手順を選び出すための解析を行う。そのために、完結フローの利用頻度を操作ログから解析する。例えば完結フローとして、図2の住所録における、追加ボタンから始まる完結フロー、閲覧ボタンから始まる完結フロー、編集ボタンから始まる完結フロー、の3つの完結フローを抽出できたものとする。これらの完結フローを以降便宜上「追加フロー」、「閲覧フロー」、「編集フロー」と呼ぶこととする。

操作ログ全体の中でのそれぞれの完結フローの出現数を求める。これらのフローのうち、出現頻度の多い完結フローを起点として、よく使われる完結フローの組み合わせを求めていく。この完結フローの組み合わせたものを複合フローと呼ぶこととする。

複合フローを求めるために、さらに操作ログの解析を行う。操作ログにおいて、起点とした完結フローの次に出現する完結フローを求める。また、合わせてその出現回数も求める。例えば、先の住所録の「追加フロー」を起点とした場合、「追加フロー」「閲覧フロー」、「追加フロー」「編集フロー」が抽出できたとする。同様に「閲覧フロー」「編集フロー」が起点となる場合の組み合わせと出現回数も求める。これらの組み合わせの中から、出現回数が少ない複合フローを除外し、更にその次に出現する完結フローを求める。以後、同様にしてさらにその後に出現する完結フローとその出現回数を求めていく。

このような複合フローを求める処理において、完結フローを求める処理を、どの段階で終了するかを判断を行う必要がある。本手法では、この処理の終了を判断する条件を、以下の3通りとする。

- (1) 同じ完結フローがもう一度出現した時
- (2) 次に出現した完結フローの出現回数が少ない時
- (3) 作成中の複合フローの出現回数が少ない時

図4に、先の住所録における複合フローの作成終了判断と、複合フローとして解析されたものの例を示す。図4における「追加」「編集」「閲覧」はそれぞれ「追加フロー」「編集フロー」「閲覧フロー」の略である。

(1)の条件は、チュートリアルとしてユーザに操作を提示すると考えた時、1つの連続する操作の中で同じ内容のものを複数回見せる必要はないと考えられるためである。また、1つの複合フローの中に同じ完結フローが何度も出現するようなものは、その複合フローの中でさらに小さな複合フローが複数回出現することが考えられる。これは、1回のチュートリアルで同じ操作手順を繰り返しユーザに対して提示することになり冗長であるため、チュートリアルとしてはふさわしくない複合フローであると考えている。例えば先の住所録での複合フローの作成において、「追加フロー」「閲覧フロー」の次に「追加フロー」もしくは「閲覧フロー」が来た場合、その時点において複合フローの作成を終了とし、この複合フローは「追加フロー」「閲覧フロー」とする。

(2)の条件は、ログ解析をした結果、ユーザの利用頻度が低いと判定されたものは、チュートリアルとして提示する候補としてはふさわしくないと考えられるためである。実際に終了処理を行う手順としては、解析を行う操作ログから抽出された完結フローの全体数から判断して、複合フローとするには候補となる完結フローが少ない場合、その時点において複合フローの作成を終了とする。例えば先の住所録での複合フローの作成において、「閲覧フロー」「編集フロー」の次の候補として、「追加フロー」が出る数が大幅に少ないとされ

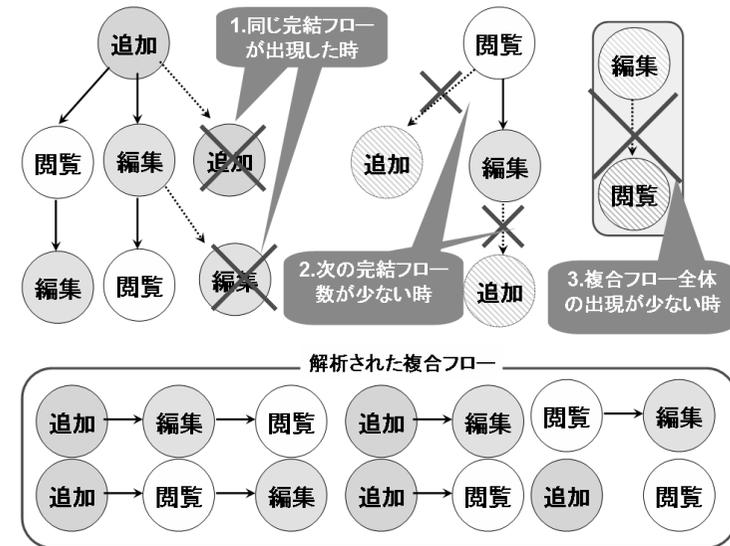


図4 操作手順解析としての複合フロー作成例
Fig.4 An example of making combine flow

た場合、この時点で複合フローの作成を終了とし、この複合フローは「閲覧フロー」「編集フロー」とする。

(3)の条件は、(2)の条件と同じく複合フローとしての出現回数が少ないものは、その複合フロー全体のユーザの利用頻度が低いと判断されるため、チュートリアルとして提示する候補としてはふさわしくないと考えられるためである。住所録の複合フローの作成例において、「編集フロー」から始まる複合フローが大幅に少ないとされた場合、「編集フロー」から始まる複合フローは候補から外すこととする。

4.4 チュートリアル対象となる完結フローの選択

4.3節の解析によって求めた複合フローのうち、出現回数の多いものをチュートリアルの候補と考える。そして、チュートリアルの候補となった複合フローに含まれている完結フローを、開発者に提示する。これは、完結フローの中にはチュートリアルとして含むことがふさわしくない、完結フローとして完全ではないものが含まれている可能性があるためである。不完全な完結フローとしては、「操作のエラーからの復帰」「キャンセルによる操作の中断」などが挙げられる。従って、このような不完全な完結フローを開発者に取り除いても

らう。

4.5 提示用データ・提示文の記述

4.4 節で選ばれた操作の流れの中には、テキストフィールドなどの文字列の入力が必要となるウィジェットが含まれていることがある。これらのウィジェットの操作をチュートリアルシステムで実現する際に、文字列の入力にはチュートリアル上で入力例が提示されることが望ましい。また、各ウィジェットの操作についての説明の文章をそれぞれの操作に付加する必要がある。この説明の文章を、提示文と呼ぶ。これらの提示用データと提示文は、開発者が記述を行う。

また、チュートリアルとなる複合フローには、ユーザに提示するための名称が必要である。しかしながら、全ての複合フローに適切な名前をつけられるとは限らない。そこで1つ1つの完結フローに、開発者が「追加フロー」のような名称をつけ、それを複合フロー内の完結フローの出現順に繋げたものを、複合フローの名称とする。この完結フローの名称を連結し、作成された複合フローの名前が、チュートリアルシステム上においてユーザが学習したい操作の選択肢として表示される。

4.6 チュートリアルシステムの生成および付加

4.3 節までに収集された操作ログの分析データと、4.4 節と 4.5 節において開発者によって選択・記述された完結フローの名称および提示用データ・提示文をもとにして、チュートリアルシステムの生成を行う。

生成されたチュートリアルシステムの動作としては、まず、操作ログの中に記述されているウィジェットの変数名を用いて、画面上の該当するウィジェットの座標を取得し、マウスポインタでそのウィジェットを指し示す処理を行う。マウスポインタは、実際のマウスポインタではなく、チュートリアルシステムで用意している擬似的なマウスポインタ(疑似マウスポインタ)を利用する。また、ボタンに対するマウスクリックや、テキストフィールドに対するテキスト入力などの代行となる処理を行う。

このようなチュートリアルシステムを、以下の手順により生成する。

- (1) 4.3 節で組み立てられたウィンドウ遷移をもとに、ウィンドウ中の各ウィジェットに対応する操作手順を組み立てる。
- (2) 組み立てた操作手順に従って、画面上のウィジェットの位置に疑似マウスポインタを表示・移動させるデモンストレーションおよび、操作に必要な各ウィジェットの操作をチュートリアルシステムに行わせる処理を生成する。疑似マウスポインタは、画像を利用することにより表示することとする。

- (3) 提示文を表示し、マウスポインタ移動のデモンストレーションに応じて提示文を切り替える処理を生成する。

これらの手順により、下記の内容を含んだ AspectJ のソースコードが生成される。

- 提示文の内容を表示するテキストエリアからなる GUI のウィンドウを作成する Java のソースコード
- ソースコードから解析したウィジェット名と、操作ログから解析し開発者によって選択された操作の流れをもとに、チュートリアル対象ソフトウェアにおけるウィジェット上に疑似マウスポインタを表示・移動させるために必要な座標情報の取得と、提示文の内容の表示を行う AspectJ のソースコード
- チュートリアルシステムのウィンドウ中における、提示文の表示内容を変更させる処理部分と、疑似マウスポインタを移動させる動作を行う Java のソースコード

4.7 チュートリアルシステムの付加

チュートリアルシステムと、学習対象となるソフトウェアはできるだけ独立に用意しておき、最終段階において組み合わせることが望ましい。そのためチュートリアルシステムは、AspectJ のソースコードとして生成する。

生成されたソースコードでは、各ウィジェットに対してポイントカットを用いて割り込むことにより、次の処理を行う。

- ウィジェットの座標を取得する処理の付加
- ボタンのクリック・テキストフィールドの文字列入力等、各ウィジェットに対応した操作を自動的に行う処理の付加
- 操作に対応する提示文を表示・切り替える処理の付加

このような生成されたチュートリアルシステムのソースコードとチュートリアルシステムを付加する対象となるソフトウェアをウィープすることによって、チュートリアルシステムを付加することができる。これによって、ソフトウェアを開発した後であっても、ソフトウェアの内容を変更することなく、チュートリアルシステムを付加することができる。また、ソフトウェアの内容を変更する必要がないため、チュートリアルシステムが不要になったとしても、ソフトウェアから外すことが容易となる。

5. 評価

本手法の評価として、レンタルビデオ業務のソフトウェアと文献⁵⁾に掲載されている通信販売のソフトウェアについて、GUI を実装し、その操作についてログを取得した。この

操作ログに対し、本手法に基づいて作成したシステムによって解析を行った。ただし、通信販売のソフトウェアの GUI は、最初にログインを促すログインウィンドウが表示されている。現時点の本手法では、最初に表示されるウィンドウをメインウィンドウであるとしているため、本評価ではログインウィンドウを除いたものを作成した。

5.1 完結フローの整合性

取得された全ログファイルのうち、どの程度の操作を完結フローとして分析できたかの評価を行った。それぞれのソフトウェアにおける本手法によって解析された完結フロー数、完結フローとして取り出されるべき実際の完結フロー数および解析されたウィンドウ遷移の数を、表 1 に示す。

表 1 解析された完結フロー数と実際の完結フロー数、ウィンドウ遷移数
Table 1 Result of unit flow analysis

	レンタルビデオ	通信販売
実際の完結フロー数	54	90
解析された完結フロー数	54	90
ウィンドウ遷移数	848	768

それぞれのソフトウェアについて、実際の操作における完結フローの数に対して、本手法によって解析することのできた完結フローは同一であった。これにより、本システムにおいて完結フローを正確に解析を行うことができることが示された。

5.2 複合フローの解析

次に、取得された全ログファイルのうち、複合フローとして解析できた操作についての評価を行った。それぞれのソフトウェアの操作ログに対して、操作ログの総操作記録数、本手法によって解析された完結フロー総数と、それらに対して複合フローとして解析された総数および、そのうち複数回現れた複合フロー数について表 2 に示す。

レンタルビデオソフトウェアの完結フローの総数 54 に対して、本手法によって解析を行った複合フローの総種類数は 11 であった。また、通信販売システムの完結フローの総数 90 に対しての複合フローの総種類数は 21 であった。

それぞれの複合フローは、同じウィンドウの遷移をたどったものをまとめることによって、エンドユーザが利用している操作の手順が多い順に並べることができる。これらのうち、同じ複合フローとして複数回現れたものは、レンタルビデオソフトウェアでは 3 種類、通信販売ソフトウェアでは 4 種類であった。複数回と評価された複合フローは、主にデータを

表 2 複合フロー解析結果

Table 2 Result of combination flow analysis

	レンタルビデオ	通信販売
総ログ操作数	2562	3758
完結フロー総数	54	90
複合フローの総数	11	21
複数回現れた複合フロー数	3	4

登録する作業を複数回行った場合と、登録後に確認を行う作業であった。このことにより、膨大な操作ログの中から、エンドユーザが行う典型的な操作の手順を解析することができると言える。

また、この複合フローにはエンドユーザの行った操作手順がソフトウェアのソースプログラムに記述されているクラスやウィジェットの名称と対応する形で収められている。これらはエンドユーザが行った操作手順を再現するために必要となるものであり、複合フローを利用することによって、チュートリアルを動作させることが可能となる。これらのことより、本手法において解析された複合フローの結果をもとにして、エンドユーザに対してよく使われる例となるソフトウェアの操作学習のサポートを行うことが可能であるといえる。

6. 関連手法

Iribe らは、初心者のソフトウェア操作学習支援のために、教師の操作を見本としてログに記録し、講習会等での初心者の操作とログを比較することで、動的に教材を作成するシステムを提案している⁶⁾。この手法では、教師の想定通りの操作を学習させることができるが、教師が想定していないユーザのソフトウェアの利用方法に対しては間違いを指摘するのみとなる。

Garcia らは、チュートリアル対象となるユーザのタスクモデルと、チュートリアルのコースを表現したシナリオを組み合わせて、チュートリアルシステムを作成する手法を提案している⁷⁾。この手法では、チュートリアルのためのシナリオと、操作を実現するためのタスクモデルとを対応付けできるように事前に準備する必要がある。

Miura らは、GUI を有する Java Applet に対して、ウィジェットの操作により発生したマウス操作の記録と再生を行う機能を付加することにより、チュートリアルシステムを作成するツールを提案している⁸⁾。この手法では、チュートリアルシステムを生成するために、ウィジェットに対して行われた全ての操作を記録し、その記録の中から意味のある操作の組

み合わせを開発者が指定をする必要がある。本手法では、ウィンドウの遷移をもとに解析を行うため、操作の組み合わせをある程度絞って開発者に提示することができ、意味のある操作を選び出すための労力を軽減している。

Eisenstein らは、インタフェースエージェントのフレームワークより、ソフトウェアのウィジェットの配置を行うツールを用いてチュートリアルに必要となる情報をエージェントに対して付加させることにより、ユーザに対して操作説明を行うチュートリアルシステムを提案している⁹⁾。本手法では、ログの内容からユーザの操作を解析し、その内容をもとにチュートリアルシステムの実現を行うことから、特定の環境に依存せず汎用的に利用できる。

7. おわりに

本研究では、ユーザの操作ログを用いて、チュートリアルシステムの自動生成および付加を行う手法について提案した。これにより、本手法によってチュートリアルシステムをソフトウェアに付加することが容易になり、ユーザに対してソフトウェアの操作方法を学習する機会をさらに増やすことができるようになる。

今後の課題としては、以下が挙げられる。

- 不要な完結フローの自動抽出
- 完結フロー・複合フロー名の操作ログからの自動抽出
- マニュアルなどと連携し、開発者の労力を軽減する手法
- 起動時の最初のウィンドウが、ログインなどのウィンドウで、ソフトウェアに存在する機能を利用する起点となるメインウィンドウではない場合の対応

参 考 文 献

- 1) M. B. Rosson and J. M. Carroll:
Usability Engineering: Scenario-Based Development of Human Computer Interaction, Morgan Kaufmann Pub., 2001.
- 2) JavaCC HOME, <https://javacc.dev.java.net/>
- 3) An Overview of AspectJ, <http://citeseer.ist.psu.edu/kiczales01overview.html>
- 4) G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, V. C. Lopes, J. M. Loingtier and J. Irwin: Aspect Oriented Programming, PARC Technical Report, 1997.
- 5) ゲリ・シュナイダー, ジェイソン・ウィンターズ著, オージス総研訳, 羽生田栄一監訳:
ユースケースの適用:実践ガイド, ピアソン・エデュケーション, 2000.
- 6) 入部百合絵, 藤原真, 安田孝美, 横井茂樹: 学習者の操作プロセスに適応した対話型ソフト

- 学習システム, 電子情報通信学会論文誌. D, 情報・システム J91-D(2) pp.269-279(2008).
- 7) G. Federico: CACTUS Automated Tutorial Course Generation for Software Applications, Proc. of the 5th International Conference on Intelligent User Interfaces(*IUI2000*), 2000.
 - 8) M. Miura and J. Tanaka: Jedemo Demonstrational Authoring Tool for Java Applets, Proc. of the 5th World Conference on Integrated Design and Process Technology (*IDPT2000*), 2000.
 - 9) J. Eisenstein and C. Rich: A GUI Editor That Generates Tutoring Agents, Proc. of the 7th International Conference on Intelligent User Interfaces(*IUI'02*), 2002.