

B-023

ソフトウェアの操作学習支援をする操作マップの自動生成手法 Automatic Generation Technique of Operation Maps for Supporting to Learn Operation Method of Software

黒岩 大輔[†]
Daisuke Kuroiwa

岩田 一[†]
Hajime Iwata

白銀 純子[‡]
Junko Shirogane

深澤 良彰[†]
Yoshiaki Fukazawa

1. はじめに

今日、多種多様なアプリケーションが開発され、それぞれの機能も増加の一步を辿っている。その一方で、アプリケーションの操作方法も複雑化する傾向にあり、エンドユーザにとってはその学習は困難になっている。そこで、ユーザの学習支援をすることがより重要になってきている。

これまで学習支援をする方法として、ペーパーマニュアル、ヘルプシステム、ナビゲーションシステム、チュートリアルシステムなどが提案され、実現されてきた。これらは指示通りに操作を進めている時の学習支援にはよいが、操作手順を誤ったり、エラーが生じた場合にはユーザは対処することが難しい。

そこでこのような場合には、ユーザが操作手順の中における自分の現在位置を理解し、今後どのように操作すればよいかを把握する事が重要である。そのために、現在表示しているウィンドウを中心とした、ウィンドウの遷移を表す地図が有効である。この地図を本手法では操作マップと呼ぶ。操作マップにより、ユーザはシステムの概要を知ることができ、自分でエラーの対策を考えることが容易になる。こうすることで学習効果がより高まるものと考えられる。

しかし、この操作マップを作成するためには、大きな労力やコストが必要となり、開発者にとっては負担となる。そこで本研究では操作マップを自動生成する手法について提案する。本研究は、現時点では Java 言語で書かれたプログラムを対象とし、GUI パッケージとしては Java Swing パッケージを前提としている。しかしこれらに限定されるものではない。本研究では、アプリケーションのソースコードを JavaML[1] を用いて解析し、その結果を用いて操作マップを生成する。JavaML とは Java ソースコードを XML 形式に変換するツールで、XML ベースの構文を使ってソフトウェア構造を直接的に表現するため、プログラミングの構造化表記に適している。

2. 操作マップとその特徴

操作マップとは、現在表示しているウィンドウを中心としたウィンドウの遷移を表す地図のことである。

操作マップでは、ある 1 つのウィンドウ (A) に着目し、下記のものを表示する。

- A を出現させるウィンドウ (A_{before})
- A_{before} において A を出現させるために必要な操作
- A から出現するウィンドウ (A_{after})
- A において A_{after} を出現させるために必要な操作

アプリケーション内におけるウィンドウ全てに対して、これら 4 点を抽出し、お互いに連結させて図示したものが操作マップである。

図 1 に操作マップの例を示す。これは Microsoft Word において、メインウィンドウから挿入する表のフォーマットを設定するウィンドウや、フォントを設定するウィンドウを出現させるために必要となる操作の経路を表している。操作マップ中の” [] ”の中の名称は、操作するボタンやメニューなどの GUI 部品 (ウィジェット) についている名前前で、” >> ”でその操作順序を示している。矢印はウィンドウの遷移を表している。

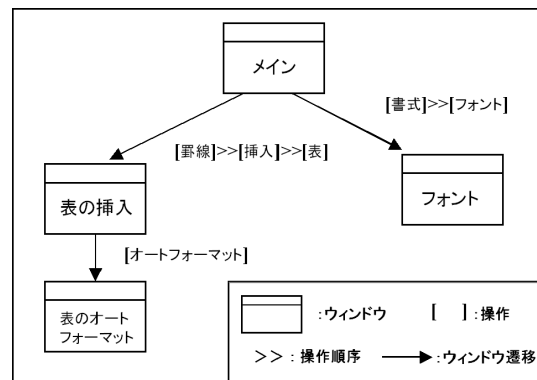


図 1: 操作マップの例

操作マップにより、現在、自分がどのウィンドウを操作しているかを把握する事ができるため、目的とする機能を達成するためには、どのような操作でウィンドウを移動すればよいのかが分かる。

操作マップを利用する大きな利点は、操作の間違いやエラーなどでマニュアルとは違った操作手順に陥った場合に、操作マップを見ることによって、その間違いやエラーなどからの復帰の手助けとなる点である。具体的には、アプリケーションにおける現在地を確認することで、間違いやエラーが起こる前の状態に自力で戻れるよう支援することである。また、複雑な遷移構造となっているウィンドウを 1 つの地図として見せることによって、アプリケーション全体の操作の流れを把握をしやすくなっている。

操作マップを利用した学習支援は、「学習」の効果的なコンセプトの 1 つである「自分で実際に経験し、解決策を考える」ということを満たす支援手法である。

3. GUI データベース

本研究でウィンドウの遷移を調べる際に必要となるデータベースを作成し、本システムに組み込んでおく。このデータベースを GUI データベースと呼び、リスナ

[†]早稲田大学大学院理工学研究科
[‡]東京女子大学

データベースとウィンドウデータベースの2つより構成する。以下に2つのデータベースの詳細を示す。

3.1 リスナデータベース

ウィンドウの遷移を抽出するために、新しいウィンドウを発生させる可能性があるウィジェット (アクションウィジェットと呼ぶ) と、可能性のないウィジェットを分類する。Java Swing パッケージで表示されているウィジェットの中で、アクションウィジェットと、それぞれのアクションウィジェットに対応したリスナを集めて、データベース化しておいたものを本研究ではリスナデータベースと呼ぶ。アクションウィジェットの例として、JButton、JMenuItem、JTextFieldなどが挙げられる。

3.2 ウィンドウデータベース

Java Swing パッケージにおいて、ウィンドウを生成する方法には大きく分けて2通りある。1つはウィジェットを宣言、初期化することによって生成する方法である。これを本手法ではオブジェクト作成型と呼ぶ。この方法で生成されるウィンドウとして、JFrameを利用したものがある。もう1つの方法として、クラスに用意されたメソッドを用いて生成する方法がある。これを本手法ではメソッド利用型と呼ぶ。この方法で生成されるウィンドウの例として、JOptionPaneやJFileChooserなどを利用したものがある。

ウィンドウを表示するメソッドもウィジェットによって決まっている。例えばウィンドウを作るウィジェットであるJFrameでは、表示するためのメソッドとしてsetVisibleを使用する。また、ファイルを選択させるウィンドウを作るウィジェットであるJFileChooserでは、表示するのにshowOpenDialog、showDialog、showSaveDialogなどのメソッドを使用する。このように、ウィンドウ作成の型名、ウィンドウを生成するウィジェット、それを表示するメソッドを集めて、データベース化したものを本研究ではウィンドウデータベースと呼ぶ。

4. 本システムの構成

本システムの概要図を図2に示す。本システムでは以下のように操作マップを自動生成する。

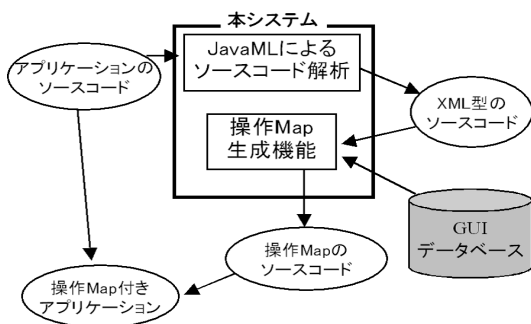


図 2: システム構成

アプリケーションのソースコードをJavaXML機能により、XMLの形に変換する。XMLに変換したソースコードより、操作マップを生成するために必要なウィンドウ情報を検索する手順は、下記の通りである。

1. ウィンドウデータベースを元に、ソースコード中でウィンドウを出現させる部分を抽出
2. 1で抽出したウィンドウ上に配置されてるウィジェットの中からアクションウィジェットを抽出
3. 2で抽出されたアクションウィジェットにおいて、リスナデータベースを元に、ユーザイベントが発生した時の処理を解析し、更にウィンドウを表示させるコードを検索
4. 3でウィンドウを表示させるコードが発見されれば、1に戻る
 - 1~4を行い、作成されるウィンドウを全て検索し、それぞれのウィンドウに対して、以下の5つの情報を整理し、ウィンドウ間の論理的なマップを生成する。

- ウィンドウのタイトル
- 別のウィンドウに遷移する際のアクションウィジェットのクラス名
- 別のウィンドウに遷移する際のアクションウィジェットのラベル名
- イベント発生がクリックで起こるのか、ダブルクリックで起こるのかというイベントの種類
- 次のウィンドウの情報を持ったオブジェクト

例として、図1の操作マップにおいて、「表の挿入」と書かれたウィンドウから、「表のオートフォーマット」と書かれたウィンドウに遷移する場合を考える。このウィンドウ遷移のきっかけとなるのは、「オートフォーマット」と書かれたボタンを左クリックすることである。この時、ウィンドウのタイトルは「表の挿入」となる。アクションウィジェットのクラス名は、ボタンがJava SwingパッケージにおいてJButtonクラスで生成されるので、「JButton」となる。また、アクションウィジェットのラベル名は「オートフォーマット」となる。そして、イベントの種類は「左クリック」となる。

最後に、その論理的なマップを元に、視覚的な操作マップのソースコードを自動生成する。

5. 終わりに

本研究では、学習支援の一助として使用する操作マップの自動生成を行う手法について提案した。今後の課題として以下のようなものが挙げられる。

- 論理マップを元にした視覚的な操作マップの実装
- 操作マップをアプリケーションと同期させる方法についての検討
- 利用頻度の高いルートを表示できるような機能の構築など、本研究で生成される操作マップをユーザにとってさらに学習しやすいものとなるような機能の追加

参考文献

- [1] Greg J. badros, JavaML Home Page
<http://www.cs.washington.edu/homes/gjb/papers/JavaML/javaml.html>