

Java ソースコードに対するアクセシビリティチェック

Accessibility Check of Java Source Code

森 崇† 岩田 一† 白銀 純子‡ 深澤 良彰†
Takashi Mori Hajime Iwata Junko Shirogane Yoshiaki Fukazawa

1.はじめに

近年のコンピュータやネットワークの普及に伴い、その利用も多岐に広がり、現状ではコンピュータは社会生活を送る上で不可欠なものになりつつある。それに伴い、健常者だけでなく、何らかの障害を持った人にとってもコンピュータ利用の必要性は大きくなってきている。

しかしコンピュータのソフトウェア、Web サイトなどは必ずしも、障害を持つ人々にとって利用しやすいものにはなっておらず、障害を持つ人々を考慮した改善が急務である。Web サイトについては、障害者にとって適切かどうかを調べるガイドライン、チェックツールなどが登場し、利用されつつあるが、アプリケーションソフトウェアの方ではそれら対策は遅れている状態である。そこで本研究では Java の Swing を用いて作られた GUI のプログラムのソースコードを解析し、障害者や高齢者にとって問題となる部分をガイドラインに基づいて見つけ出すためのチェックツールについて提案する。

2.アクセシビリティ

アクセシビリティとは、障害者や高齢者であってもそのソフトウェアや Web サイトをどれほど問題なく利用できるか、ということの意味する。ソフトウェアや Web サイトを利用する際に、問題が少なく操作しやすい、利用しやすいということであれば、そのソフトウェアや Web サイトはアクセシビリティが高いと言える。

例えば、現在では GUI を用いたソフトウェアが主流となっているが、マウスによる画面上のボタンのクリックなどの視覚的な操作が必要で、特に盲目の人にとっては利用することが難しい。また弱視や色弱の人にとっては、画面上の小さな文字、色付きの部分を正確に認識できない等が考えられる。アクセシビリティ向上のためにはキーボードによる各操作の実現、スクリーンリーダーや点字ディスプレイへの対応、画面上の色調やコントラスト、文字の大きさなどの適切な設定、といったことが必要となる。

2.1 ガイドライン

現在のアクセシビリティの取り組みについては、アメリカのリハビリテーション法[1]や W3C Web Contents Accessibility Guideline (WCAG)[2]が広く知られている。

リハビリテーション法はアメリカ政府が制定したものであり、その中の 508 条では、政府機関が購入する IT 機器・ソフトウェアは、障害者などにも利用できるものでなければならないとされている。

WCAG は、どうすれば Web コンテンツが障害のある人にも利用しやすいものになるか説明している。現在、企業や団体などで策定されているガイドラインの多くがこの

WCAG を元に行っている。このように、現在ではアクセシビリティの重要度が認知され、アクセシビリティの高い製品の作成が不可欠になってきている。

2.2 Java Accessibility API

Java Swing には、GUI コンポーネントに対するアクセシビリティ向上のための javax.accessibility パッケージ[3]がある。これはコンポーネントとコンポーネントアクセスを提供するアシスティブテクノロジー間の結びつきを決めている。例えばこの API を用いることで、Swing のソフトウェアのクリーンリーダー・点字ディスプレイへの対応が可能となる。従ってアクセシビリティの高い Java ソフトウェアの作成のためには、この API を利用することになる。

3.研究方針

本研究では、ガイドラインに基づいて Java ソフトウェアのソースコードを解析し、アクセシビリティ上問題があると考えられる部分を開発者に提示する事を目指す。なお本システムは、eclipse のプラグインとして動作させる。本システムのアーキテクチャを図 1 に示す。本システムでは Java ソフトウェアのソースコードを JavaCC[4]により構文解析し、その結果を本システムに読み込むことで、アクセシビリティのチェックを行う。

3.1 JavaCC でのソースコード変換

JavaCC とは Java Compiler Compiler の略であり、字句解析や構文解析を行うクラスのソースコードを自動生成するツールである。まずこの JavaCC によって、ソースコードをチェックに適した形に変換する。具体的には、ソースコードから各ウィジェットの変数の型や名前、使用されているメソッド名等を抽出する。そしてそれを整理し、どのような名前のウィジェットにどのようなメソッドが指定されているかを一覧にまとめる。

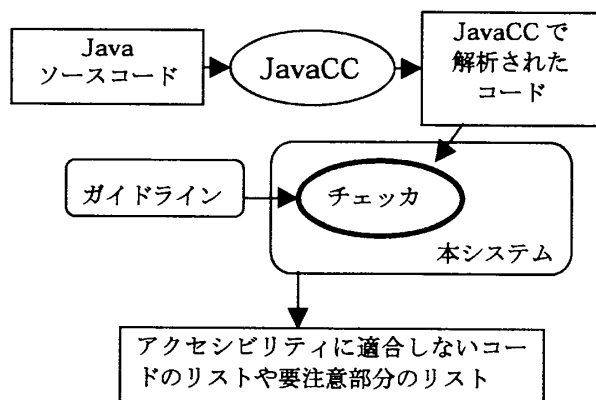


図1: ソースコード解析、チェックの流れ

† 早稲田大学大学院理工学研究科

‡ 東京女子大学現代文化学部

表1：アクセシビリティチェックのガイドラインの一例

ガイドライン	ガイドラインの説明	ソースコードにおけるチェック項目
部品に記述をつける	スクリーンリーダなどでは、この内容を読み上げる	setAccessibleName()メソッド setAccessibleDescription()メソッド
キーボード操作可にする	視覚、上肢障害の人の操作	setMnemonic()メソッド FocusTraversalPolicy クラス
文字や背景の色彩に注意する	色弱の人でも見やすい、認識しやすい色を付ける	setForeground()メソッド setBackground()メソッド

3.2 チェック項目

本システムでは、用意されたガイドラインに基づいて、対象となるソースコードのアクセシビリティのチェックを行う。その際のソースコード中のチェック個所の例を、表1に示す。本研究で利用するガイドラインは様々な企業、団体などで定められているガイドラインをまとめ、ソフトウェアとして必要なものを集めたものである。表1はその一例である。

まずスクリーンリーダへの対応のため、ウィジェットにテキストの情報を与えておくことが重要である。これは部品に対して名前や説明文などを与えるメソッドがチェック項目となる。

次にマウスを使えない場合のためにキーボード操作に対応するということが重要である。これは各ボタン等にニーモニック(Alt キーなどと組み合わせるショートカットキー)を設定したり、Tab キーによるフォーカス順を設定する部分がチェック項目となる。

また色弱の人にも見やすいように、色・コントラストにも注意を払う必要がある。これはウィジェットの色を設定するメソッドがチェック項目となる。

ニックが設定されていないため、最下部のような結果を表示する。

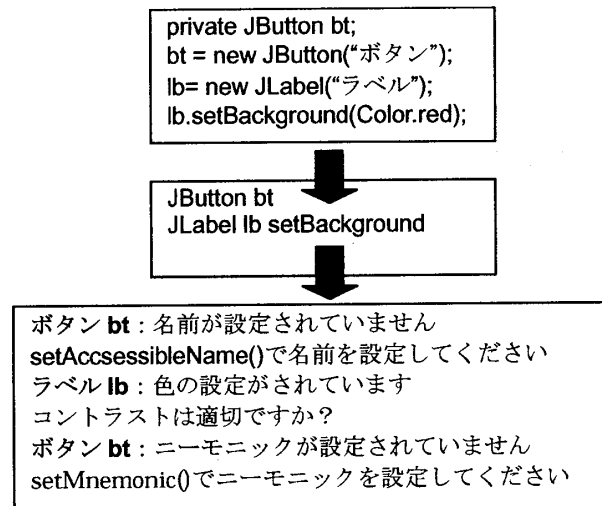


図2：チェック例

3.3 チェック結果

3.2 のチェック項目に基づいてソースコードを解析し、アクセシビリティに適合しない部分や、注意を喚起すべき部分を指摘する。

例えば、表1のガイドラインに従えば、プログラム中でJButtonを使った際は、setAccessibleName()メソッドやsetAccessibleDescription()メソッドによりテキスト情報を与える必要があるが、ソースコードから情報を解析した結果、その記述が無ければ、記述をするように開発者に指摘をするということになる。

また、ウィジェットの色を使い方などはアクセシビリティに適合する、しないというのはいかに言えないが、上記のJButtonの時と同様に開発者に色の指定をしている部分を指摘し、色彩、コントラストが適切なものになっているかどうか、改めて考えてもらうように促すということになる。

4. 実行例

図2にチェックの実行例を示す。最上部のソースコード(一部)を本システムで読み込むと、中央部のような情報を抽出する。この情報は、GUI中で使われているウィジェットのクラス名と変数名、そしてどのようなメソッドが使われているかというものである。この情報から、JButtonに対してはアクセシビリティのための名前や説明文、ニーモ

5. 終わりに

本研究では、ガイドラインに基づいてJavaのソフトウェアのアクセシビリティをチェックする手法について提案した。今後の課題は以下のとおりである。

- 本システムにおけるアクセシビリティチェック妥当性の検証
- 本システムの実装と評価
- 開発者独自のアクセシビリティのガイドラインを取り込む機構についての提案

参考文献

- [1]508条について：
<http://www.udit-jp.com/Section508/508/what.html>
- [2]W3C Web Content Accessibility Guidelines：
<http://www.w3.org/TR/WAI-WEBCONTENT>
- [3]javax.accessibility (Java 2 Platform SE 5.0)
<http://java.sun.com/j2se/1.5.0/ja/docs/ja/api/javax/accessibility/package-summary.html>
- [4]JavaCC コンパイラ・コンパイラ for Java：
五月女健治 著 テクノプレス 2003年