

# 課題の解としての GUI プログラムの評価システム

横濱 彰則, 海谷 治彦, 海尻 賢二

380-8553 長野市若里 4-17-1 信州大学工学部 情報工学科

## 1、はじめに

プログラミング教育において、実行による動的なテストは効果的であり、CUI プログラムにはよく用いられる手法であるが、GUI プログラムを対象とした実行テスト体系はない。そこで本研究は、「評価者の負担の軽減」「学習者への効果的なフィードバック」を主眼に置き、Java 環境下において、GUI プログラムの評価を簡便に行うことを可能にするシステムの開発を目的とする。

## 2、概要

### 2.1、本研究の動機

GUI プログラムを動的テストにかけようとした場合、JUnit[1]や JFCUnit[2]を用いる方法がある。しかし、これらは回帰テストのフレームワークのため、同じプログラムを複数回テストすることには向いているが、学生のプログラム課題提出物のような、「実行結果は同じであるが、コーディングスタイルがそれぞれ違うプログラム」を複数個テストすることには向いていない。そこで本研究は、提出されたさまざまなバリエーションを持つプログラムに対し、動的テストを実行できる仕組みを考え、そのシステムを作することを目的とする。

### 2.2、本手法の基本概念

本研究の核となるものは、「テストツール」と呼ばれる、提出物の動的テストを実行するプログラムである。テストツールは、予め評価者が用意しておいた“テスト内容を指示するデータファイル”（以下、「テスト系列データ」と呼ぶ）に基づいてテストを行い、評価を下す。たいていの場合、講義には複数回課題が出されるはずであるが、本手法では、テスト系列データを差し替えるだけで、それぞれの課題への対応が可能となっている。テスト系列データは、提出されたプログラムへのテスト内容が指示してある XML ファイルであり、XML タグの一つ一つがプログラムへのテスト命令と対応している。た

例えば、「push タグ」は「Button を押せ」、「assert タグ」は「Label」の「内容を照合する」などである。これらの命令に基づいて、テストツールは、マウスイベントの生成や、プログラムの遷移状態の取得を行う。実際のイベント生成や取得は、ReflectionAPI を用いて行う。ReflectionAPI は、Java に用意されている機能で、任意のクラスのメソッドやフィールドを実行時にアクセスできる仕組みである。なお、テスト対象となるプログラムは Java の標準的な GUI コンポーネント (Swing や awt) さえ使っていればどのような書き方をしても良い。また、テストするために特別なコードを付け足す必要がないため、学習者にとってはテストがしやすいものとなっている。

### 2.3、テスト系列データの生成

テスト系列データは XML で書かれており、プログラムからは読み書きしやすい形式となっている。しかし、プログラムのテストは実行する項目が多いため、それに比例して、タグが多く、あるいは構造が複雑になり、人間では管理しきれなくなる。また、複数個の課題のテストに対応するためにも、このテスト系列データを如何に簡便に迅速に作れるかがシステムとして重要となる。そのため、本手法ではテスト系列データを作る「ウィザード」を用意することにした。評価者がウィザードに対して操作を行うことでテスト系列データを半自動的に作り、評価環境作りの負担を軽減することを目指す。

## 3、ウィザードの仕様

ウィザードは、テスト系列データの作成の支援機能を有し、対話的にテスト系列データを作成できることを目指しており、プログラムがどのようなコンポーネントを持っているか調査する機能を持っている。そのため、部品を列挙して各々の部品に対し、評価者が細かいテストを指定することが可能である。また、GUI プログラムでは、あるコンポーネントに対し操作を行った場合、別のコンポーネントの状態が変わるというものが程度定型的に決まっているものがある。そこで、ウィザードにそのような組み合わせを提示させることもできる。さらに、ウィ

ザードは「リプレイ機能」を持っている。リプレイ機能は、模範解答となるプログラムを用意し、それをウィザード上で実行することで、テスト系列データを作成する機能である。評価者はテストの手順を模範解答プログラム上で操作し、ウィザードはその挙動と状態遷移を逐一記録する。評価者の操作が終了したところで、ウィザードは記録したデータを、評価者に対話的に提示する。記録した行動がすべてテストに関係あるとは限らないため、評価者はそれらの中から必要なものを取捨選択する。指示されたものを用いて、ウィザードはテスト系列データを作成する。課題のテスト時に、評価者が行ったテストを再現するような形になるため、リプレイ機能と呼ばれる。

#### 4、テストツールの仕様

本手法は、どのようなコーディングスタイルのプログラムでもテストできるようにすることを目指しているが、まずは、本大学の講義の教科書としても使われており、また、初心者向けの GUI 部品が程よく使われたプログラムである、SunMicroSystems の公式サイト内の Swing のチュートリアルページ[3]で示したプログラム群をテストできるようにすることを本研究の目標とした。このページのプログラムはとてもしっかりやすく模範的なコードであり、また、基本的なコンポーネントの組み合わせでできているプログラムなので、テストツールの検証に最適である。

##### 4.1、テスト手法の詳細

提出する課題は、jar アーカイブで提出する必要がある。また、テストツール自体も、Java で構築されている。テストツールは提出されたプログラムを ReflectionAPI により、動的にインスタンス化する。ReflectionAPI は jar アーカイブ内の任意のクラスを、動的に文字列で指定しインスタンス化することができるため、課題やテストツールのコードを修正することなくテストが可能である。その後、テストツールは、awt や Swing に用意されている contentPane を通して、GUI オブジェクトとアクセスを行う。contentPane は GUI 部品の配置や状態を保持しているオブジェクトであり、内部に個々の部品のインスタンスをツリー状に保持している。

##### 4.2、テストツールの稼働

上記の手法を用いたテストツールは、WBT として学習者に提供する。そのために、HTTP サーバを用意し、この上でテストツールを実行させる。

テスト用サーバは、テストツールと同じ、Java でできている Servlet を用いている。まず、評価者は、用意したテスト系列データをテスト用サーバにセットしておく。学習者にはインターフェイスとして、Web アプリケーションを用意し、学習者はサーバに対し課題をアップロードする。サーバはテスト系列データに基づき、課題を動的テストにかけ、その過程と結果を学習者に対してフィードバックする。

#### 5、本手法の制限

本手法では、Swing や awt を用いたいわゆる一般的な Java の GUI プログラムであれば何でも評価することが可能である。現状、テストツールがテスト対象とする GUI 部品は、ボタンやラベルなどのごく基本的なものばかりであるが、多少の改良でもっと多くの GUI 部品をテスト対象とすることができる。ただし、本ツールは回帰テストのためのフレームワークではないため、すべての可能性に対応するようなことはしない。また、現状のテストツールではテストすべき GUI コンポーネントを一意に識別できないため、学習者はコンポーネントに対して既定の名前をつけてやる必要がある。付けなかった場合、プログラムが正しくてもテストに失敗する。これに関しては、コンポーネントの意味や役割を推測するような仕組みをテストツールに盛り込むことで対応する。なお、コンポーネントに因らない出力（例えば、コンポーネントに図形を描画するなど）は現状では認識できないが、人間の起こしたイベントだけでなく、プログラムのプロファイルを解析する（どのクラスのどのメソッドが呼ばれたか等を記録する）ことで対応することが可能と思われる。

#### 6、まとめ

本システムにより、評価者と学習者双方の負担を軽減し、また素早いフィードバックを行うことができる。手法としては、JUnit や、JFCUnit などと似ているが、学習者も評価者もテスト専用のコードをプログラムに書き込む手間がないのが本手法の強みといえる。

#### 参考文献

- [1] JUnit, <http://junit.org/index.htm>
- [2] JFCUnit, <http://jfcunit.sourceforge.net/>
- [3] Creating a GUI with JFC/Swing, <http://java.sun.com/docs/books/tutorial/uiswing/index.html>