

Title	例示によるプログラミング手法を用いたGUIテストケーススクリプト自動生成
Author(s)	筒井, 圭一朗
Citation	
Issue Date	2003-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1674">http://hdl.handle.net/10119/1674</a>
Rights	
Description	権藤克彦, 情報科学研究科, 修士

# Automatic Generation of Test Case Scripts for GUIs using "Programming By Demonstration"

Keiichiro Tsutsui (110082)

School of Information Science,  
Japan Advanced Institute of Science and Technology

February 14, 2003

**Keywords:** Programming By Demonstration, Generalized reasoning, Automatic testing, Automatic generation, PBD, GUI.

## 1 Background

The widespread use of GUIs is leading to the construction of more and more complex GUIs. With the growing complexity the cost for GUIs testing is growing in a huge.

Most of the automatic testing tools succeed to reduce the time of test case execution, but not of test case generation[2, 3, 4].

Generally speaking, GUIs are difficult to automatically generate test cases[7], because GUI's test cases are likely to be very large due to *event driven mechanism*, *many static/dynamic properties in GUI objects*, and *many combinations of operation patterns*. Moreover, it is difficult to select useful test cases from the large number of ones, because human, not computer, needs to check one by one if what you get in the display is what you want or not.

Today, research on Programming By Demonstration(PBD) is attracting [1, 6]. PBD is a technique to automatically construct a program by recognizing user's GUI operation patterns. In a PBD system, users execute operations on examples and the system constructs a general program by extracting some patterns from a sequence of the operations. Thus a PBD system allows users to construct programs without programming.

Our basic idea is to apply PBD to automatically generate test case scripts, where testers execute operations on test cases and the system constructs a reusable test case script with high coverage.

## 2 Purposes

Our purpose is to provide a PBD algorithm that generates useful test cases from tester's operation sequence, to develop the test cases generator using the algorithm and to evaluate the usefulness.

The automatic generator generates two kinds of test case scripts:

- Generalized Test Case Scripts

The generator reasons general structure from an operation sequence which is just time-series, and generates test cases that are more readable, more reusable, and easier for testers to edit.

- Fluctuated Test Case Scripts

The generator considers a tester's operation sequence as typical test case, and generates the high coverage test cases by giving the operation variation.

Thus our goal is to automatically generate reusable test cases with high coverage only from tester's operations on applications.

Moreover, we consider a GUI automatic test case execution system to detect GUI errors on layout and appearance, and show the construction guideline for the test case execution environment. We actually develop the system, and evaluate it.

## 3 Approach

To apply PBD to a GUI test case automatic generator, it is necessary for the generator to recognize and extract some patterns from GUI operation sequence by a tester, which have useful characteristics to generate test cases.

In this research, our generator divides operation sequences into two patterns. One is an operation sequence in the same window; the other is an

operation sequence including window changes. This is because functions in most GUI systems are hierarchically designed in the sense that each function is deployed in one window, users execute operations on the functional hierarchy and testers design useful test cases by examining the functional hierarchy.

And the generator generates two kinds of test case scripts using these characteristics.

- **Generating generalized Test Case Scripts**

The generator reasons and generalizes a given operation sequence using pattern's classification and frequency, and produce a script with control structures, sub-routines and some comments for readability and reusability. This is done by heuristics rules.

- **Generating fluctuated Test Case Scripts**

The generator considers a tester's operation sequence as typical test case, and generates the high coverage test cases by applying the template. The template is described about the operation plan to generate test case variation.

We conduct an experiment on automatic generation and evaluate whether these methods generate useful test case scripts from a given GUI operation sequence.

## **4 Experiment Evaluation**

We developed the GUI test case automatic generator that we proposed, and conducted a generation experiment on some small applications.

- **Generation experiment for generalized Test Case Scripts**

L-pattern's or T-pattern's scope was clear by sub-routine-izing, and repeat performing part was clear by control structures. So we checked improvement in readability and reusability.

- **Generation experiment for fluctuated Test Case Scripts**

The fluctuated script was generated from the original operation sequence with following the test operation plan of a template. We

checked that event generating was 1.5 ~ 3 times as much as the original operation sequence.

These show that the usefulness of the generator which automatically generates test cases only by operating application.

We developed the environment which takes screen-shots per event. We set GUI errors that commented out the handlers of some widgets and the environment could detect the GUI errors.

Some important lessons demonstrated in this research:

- Many sub-routine was generated because of their granular is too fine.
- It is not usefulness that our generator do not generate control structure about T-pattern.
- We could not evaluate the usefulness of fluctuated test cases because of the difficulty to evaluate the coverage.

## 5 Conclusions

In this research, we focus that most of the automatic testing tools succeed to reduce the time of test case execution, but not of test case generation. And represented the PBD algorithm which generates useful test cases.

This algorithm generates generalized test cases script and fluctuated test case scripts , by extracting the pattern feature from GUI operation sequence, reasoning for generality and applying a template.

We obtained three results.

- Script code which match our reasoning indicates readability and reusability.
- Our reasoning is not effectiveness for code which do not match it.
- It is difficult to evaluate the usefulness of fluctuated test cases.

There was little information to consider the problems because of conduct a generation experiment on some small applications. In the future we plan to examine the problems on some large ones.

Finally, we discussed the improving point of the generalized reasoning, and how to set screen-shot as GUI test oracle and problem of the test logging timing. So we showed the difficulty to automate GUI Testing.

## References

- [1] Allen Cypher, Watch What I Do: Programming by Demonstration, <http://www.acypher.com/wwid/>
- [2] Atif M. Memon, Martha E. Pollack, Mary Lou Soffa, Using a Goal-driven Approach to Generate Test Cases for GUIs, ACM 1999
- [3] Atif M. Memon, Martha E. Pollack, Mary Lou Soffa, Automated Test Oracles for GUIs, ACM 2000
- [4] Atif M. Memon, Martha E. Pollack, Mary Lou Soffa, Hierarchical GUI Test Case Generation Using Automated Planning, 2001
- [5] Cameron Laird, Kathryn Soraiz, Testing GUI applications, <http://www.itworld.com/AppDev/1262/UIR010316testinggui1/> Unix Insider 2001
- [6] Myers B.A., Demonstrational Interfaces: A Step Beyond Direct Manipulation, IEEE Computer vol.25 No8 pp.61-73 1992
- [7] Paul Gerrard, Testing GUI Applications, EuroSTAR '97, 24-28, 1997