第 35 卷 第 3 期 计 算 机 工 程 2009 年 2 月
Vol.35 No.3 Computer Engineering February 2009
·软件技术与数据库· 文章编号：1000—3428(2009)03—0094—02 文献标识码：A 中图分类号：N945

基于 HFSM 模型的 GUI 自动测试

姜文君[1]，李建涛[2]
(1. 湖南信息科学职业学院计算机科学与技术系，长沙 410100；2. 三一重工，长沙 410100)

摘 要：针对图形用户界面(GUI)软件的层次性，采用分层有限自动机(HFSM)模型对被测 GUI 软件建模，将 GUI 测试转换为预期 HFSM 和实际 HFSM 的一致性测试。模型自动搜索 GUI 控件，提供接口，采用均匀设计思想自动产生测试数据，由测试数据驱动测试执行。根据 GUI 软件的特点，采用增量方式进行模型构造和测试，从而避免错误积累。

关键词：GUI 自动测试；分层有限自动机；数据驱动；增量测试

Automatic GUI Test Based on HFSM Model

JIANG Wen-jun[1], LI Jian-tao[2]
(1. Department of Computer Science and Technology, Hunan Vocational College of Information and Science, Changsha 410100;
2. SANY Company, Changsha 410100)

【Abstract】This paper focuses on Graphical User Interface(GUI)'s automatic test. The most difficult thing is test case generation. This paper adopts a new technique to test GUIs automatically. It describes GUIs with Hierarchical Finite State Machine(HFSM), and then the work is converted into the consistency testing of the expected HFSM and the actual HFSM. It searches GUIs' controls initiatively and automatically, and generates the tesing data with 'uniform design' method. Testing data drive the test to be executed. According to the characteristics of GUIs, this paper adopts incremental mode to construct the model and to do test, to avoid error accumulation.

【Key words】automatic Graphical User Interface(GUI) test; Hierarchical Finite State Machine(HFSM); data driving; incremental test

1 概述

现有的图形用户界面(Graphical User Interface, GUI)软件自动测试技术有以下不足：录制/回放技术[1]过程繁琐，脚本的界面依赖性强；以关键字驱动思想为核心的软件自动化框架支持[2](Software Automation Framework Support, SAFS)框架自动化程度低；基于结构化面向对象形式语言(Structured Object-oriented Formal Language, SOFL)规约的测试用例生成模型[3]，规格说明书要用 SOFL 语言进行描述，实用性差；PATHS(Planning Assisted Tester for graphical user interface Systems, PATHS)系统[4]实现了自动产生测试路径，但其算子需要人工指定前提和结果。

本文提出一种数据驱动的分层有限自动机(Hierarchical Finite State Machine, HFSM)模型，将 GUI 软件测试划分为窗体内和窗体间 2 个部分，并分别探讨测试算法。本文为 GUI 软件增量式地生成预期 HFSM 和实际 HFSM，并实时比较两者的一致性。基于本文思想开发的分层 GUI 自动测试(Hierarchical GUI auto-testing, HGAT)工具，使用简单，自动化程度高，测试用例可重用性好。

2 HFSM 模型与 GUI 软件自动测试

2.1 相关定义

定义 1 分层有限自动机(HFSM)

HFSM 是一个五元组 $M=(I,O,S,E,\delta)$，其中，I, O, S 分别是输入符号、输出符号和状态的有限非空集合；S 可以是(窗体内部变迁引起的)一般状态或(窗体间变迁引起的)复合状态；E:{e|e=(*input*, *procedure*, *output*)}是事件集，*input*, *procedure*, *output* 分别表示一个事件的输入、处理和输出；$\delta: S\times E\rightarrow S$ 是状态变迁函数。本文中的“层”是针对 GUI 软件的执行而定义的，不同层次中的状态没有质的差异。实质是针对 GUI 软件的执行分片和分类进行测试。

定义 2 可输入控件

可输入控件是指能够接收用户键盘输入或鼠标选择数据的 GUI 控件。

2.2 模型结构

模型结构如图 1 所示。

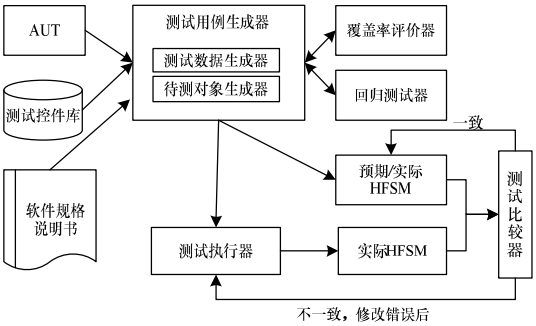

图 1 模型结构

其中，测试用例生成器是核心。它接收被测软件(Application Under Test, AUT)，并根据测试控件库和软件规格说明书，为其生成测试用例。

测试控件库是测试执行的基础。每个在 GUI 控件内发生的动作，实际上都调用了背后应用程序的内部函数。自动化

作者简介：姜文君(1982－)，女，讲师、硕士，主研方向：软件测试；李建涛，工程师、硕士

收稿日期：2008-05-10 **E-mail：**wenjj8a@163.com

测试旨在模拟人类在 GUI 上的鼠标键盘动作，关键是开发一个测试控件库，用它来创建函数完成 GUI 部件的完整搜索，为找到的特定部件编写测试脚本，并完成预期的人类动作。本文利用 User32.dll 来开发测试控件库。

User32 是一个内存映射的 dll，它被所有正在运行的进程所共享。利用这一点，调用 API 函数的应用程序就能在所有活动地址空间可见，从而能被测试工具识别。另外，能够识别 GUI 的函数也位于 User32.dll。通过调用一个或多个 User32.dll 函数，定义好 GUI 上出现控件的处理函数，存放在测试控件库中，供测试执行时自动调用，从而能自动识别 GUI 上的控件并自动执行其动作。测试控件库还可重用和扩展。在测试之前，要保证 GUI 上的控件在测试库里都有相应的处理。覆盖率评价器计算测试用例的覆盖率，若覆盖率不高，可重新生成测试用例，直到达到满意的覆盖率。回归测试器负责对软件回归测试进行处理。

3 基于 HFSM 模型的 GUI 自动测试

3.1 窗体内测试

窗体内测试(*execute*(*W*))的基本步骤如下：

(1)确定控件信息：利用测试控件库里的处理函数，自动确定该窗体的内部控件及其属性动作信息。

(2)确定测试数据：自动生成该窗体的均匀设计表，作为可输入控件的测试数据。

(3)确定待测对象：测试人员指定待验证的属性、域或事件，HGAT 自动将其作为测试数据存放在专门的 XML 文件中，同时自动生成预期模型(HFSM)的窗体内脚本。

(4)自动运行脚本：HGAT 利用测试数据驱动测试脚本的运行，此时生成临时的实际模型(*thfsm*)，当 *thfsm* 与 HFSM 不符时，认为是错误，测试人员可及时修改，直到二者一致(*thfsm* 就记为 *hfsm*)。

窗体内的自动测试流程如图 2 所示，这样就完成了该窗体以控件为单位的单元测试和窗体集成测试，同时建立了部分 HFSM 模型，并使该模型的预期和实际保持一致。该过程突出的好处是能够从规格说明书开始发现软件设计缺陷。

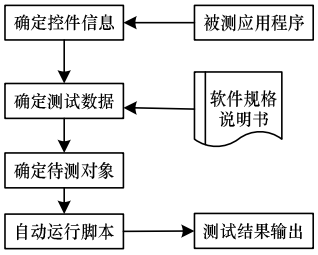

图 2　窗体内自动测试流程

3.2 窗体间测试

窗体间测试不用另外搜集数据，只需执行窗体内测试的步骤(3)和步骤(4)，HFSM 的窗体间部分也是通过指定预期输出自动生成的，窗体间测试即是验证 *hfsm* 与 HFSM 窗体间变迁的一致性。

窗体间的测试算法如下：

算法 1

void Test(window W_{mn})

{ /*对窗体 W_{mn} 进行窗体内测试，记录新产生的窗体，并增加相应的状态及状态变迁 */

Execute(W_{mn});

Record new windows $W_{m+1\,1}$,…, $W_{m+1\,k}$;

for(i∈[1, k], i++){

if (not Contain $W_{m+1\,i}$ in thfsm)

Add State $W_{m+1\,i}$ to thfsm;

Add Transition (W_{mn}, $W_{m+1\,i}$) to thfsm;}

若 thfsm 与 HFSM 此部分相符合，

则通过测试，hfsm = thfsm；

否则，更改错误后，重复上面过程。}

3.3 完整测试过程

HGAT 测试 AUT 的完整过程如下：(1)对 AUT 的所有窗体进行窗体内测试；(2)从待测软件的启动窗体开始，以指定的方法激发按钮和菜单事件，直到所有窗体跳转事件都能正确响应(见算法 1)。

增量式构造及测试算法如下：

算法 2

void　IncrementalTest(AUT)

{Test(W_1);

do{for (each $W_{i\,j}$)

Test($W_{i\,j}$);

}While(has new window or transition);

/*一个特殊状态—终止状态，每个终止程序的事件都能从当前状态到达终止状态*/

whenever quitAUTevents happens,

Add Transition(CurrentState, StopState)to hfsm; }

HGAT 构造的 HFSM 并不是一个简单树形结构，而是存在下层到上层、同层之间的变迁。它的整体形状如图 3 所示。将每个复合状态替换为一般状态，即构成窗体内子 FSM。

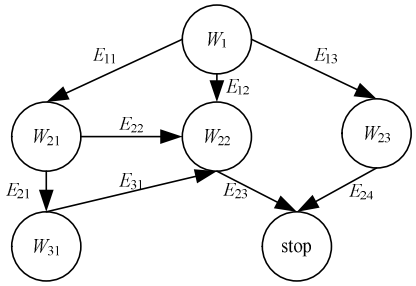

图 3　GUI 软件的 HFSM 描述

3.4 路径覆盖率

本文定义 2 个路径覆盖率作为实际测试时的覆盖率标准，其公式如下：

$$intraWindowCoverage=UnifyTable\times ControlList|/|Input\times ControlList| \quad (1)$$

$$interWindowCoverage=|executeTransitEvents|/|designateTransitEvents| \quad (2)$$

由式(1)可见，均匀设计表的质量直接决定了窗体内测试覆盖率的高低。式(2)体现了本文模型的一个特点：所要即所得。测试人员指定想要测试的对象(域、属性、事件)，测试脚本就生成相关信息。在软件无变化的情况下进行回归测试，窗体间路径覆盖率总是 100%。

4 实例与评价

本文通过一个实例来说明 HGAT 系统的测试过程。以简化的用户登录界面为测试对象，该界面元素如下：2 个 label 控件：lblUserID 和 lblPassword，2 个 textbox 控件：txtUser 和 txtPassword，2 个 button 控件：btnLogin 和 btnCancel。运行环境：CPU(Intel Pentium Ⅲ 800)，内存(256 MB)，硬盘(40 GB)，操作系统(Windows XP)。HGAT 自动测试过程及耗时情况如表 1 所示。HGAT 采用增量式方式处理回归测试，如表 2 所示。

(下转第 129 页)