文章编号: 0438-0479(2002)06-0706-05

一种 VFSM 的图形用户界面的测试方法

陈启安[1], 李小将[2], 李蜀瑜[2]
(1. 厦门大学计算机科学系, 福建 厦门 361005; 2. 西北工业大学计算机科学与工程系, 陕西 西安 710072)

摘要: 采用有限状态机(FSM)来模拟GUI的测试问题具有很大的局限性. 作者针对基于FSM的GUI测试的局限性, 提出了一个变量有限状态机(VFSM)的形式化模型, 该VFSM通过引入一组变量, 可以使用较少的状态来模拟复杂的GUI, 并且根据GUI设计规范来构造VFSM相对比较简单; 然后, 给出了一个VFSM->FSM的转化定理和转化算法; 接着介绍了基于VFSM的GUI测试方法的步骤; 最后给出了该方法的一个应用实例数据, 数据表明采用VFSM的GUI测试方法大大减少了状态、变换和输出的数量.

关键词: 有限状态机; 变量有限状态机; GUI; GUI测试; 软件测试; 用户界面
中图分类号: TP 316.2 文献标识码: A

图形用户界面(GUI: Graphic User Interface)由于拥有大量的状态、基于交互和事件驱动的输入以及复杂的图形输出界面, 使得GUI测试成为当前研究的难点. 有限状态机(FSM: Finite State Machines)可以部分地模拟GUI的测试问题. 对一个给定的GUI软件P, 根据给定软件的GUI设计规范构造一个有限状态机FSMp, 根据FSMp生成测试用例(关于基于FSM测试用例生成方法[1]), 实现图形用户界面软件的测试. 然而, 采用FSM的GUI测试方法具有很大的局限性: 1) 采用FSM来模拟复杂的GUI, 需要使用大量的简单状态, 其中很多状态是冗余的; 2) 根据GUI设计规范构造一个FSM具有相当大的难度. 论文首先针对基于FSM的GUI测试的局限性, 提出了一个变量有限状态机(VFSM: Variable FSM)的形式化模型, 该VFSM通过引入一组变量, 可以使用较少的状态来模拟复杂的GUI, 并且根据GUI设计规范来构造VFSM相对比较简单; 然后, 给出了一个VFSM→FSM的转化定理和转化算法; 接着介绍了基于VFSM的GUI测试方法的步骤; 最后给出了该方法的一个应用实例数据和结论.

1 FSM 和 VFSM 的定义

下面给出FSM和VFSM的形式化定义, 并举例说明.

定义 1[2] 一个有限状态机(FSM) M 用一个五元组表示如下:

$M = (I, O, S, \delta, \lambda)$

其中:

I: 表示非空的有限输入集合;

O: 表示非空的有限输出集合;

S: 表示非空的有限状态集合;

$\delta: S \times I \rightarrow S$ 的状态转换函数;

$\lambda: S \times I \rightarrow O$ 的输出函数.

设有限状态机的当前状态 $s \in S$, 如果接收到一个输入 $a \in I$, 有限状态机转向的下一个状态用 $\delta(s, a)$ 表示, 产生的输出用 $\lambda(s, a)$ 表示.

有限状态机可以用状态转换图和状态表表示. 状态转换图是一个有向图, 图中的节点表示有限状态机的状态, 状态图中的边线表示状态的转换, 每条边标有状态转换的输入和输出. 图1是一个有限状态机的状态转换图, 假设当前的状态为 s_1, 对于输入 b, 状态机转到状态 s_2, 并且输出1. 该有限状态机用状态表表示如表1所示, 每行表示一个状态, 每列表示一个输入符号, 每行的状态和每列的输入决定了下一个状态和输出.

一个变量状态有限机(VFSM)和上述的有限状态机相似, 但是引入了一些全局变量, 每个变量在执行输入序列的过程中指定了一组值, 用来确定下一

收稿日期: 2002-04-25
作者简介: 陈启安(1962-), 男, 副教授

个状态和输出. 同时, 进行状态转换时可以修改变量的值. 因此, 通过变量作为状态转换的因素, 一个 VFSM 可以使用比 FSM 更简单的结构和更少的状态模拟实际界面.

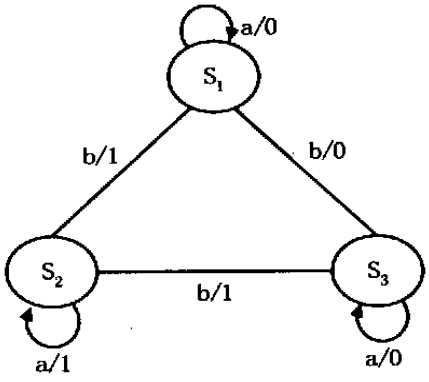

图 1　有限状态机的状态变换图

Fig. 1　State transformed chart of limited state machine

表 1　图 1 所示有限状态机的状态表

Tab. 1　State list of limited state machine in Fig. 1

	a	b
S_1	S_1, 0	S_2, 1
S_2	S_2, 1	S_2, 1
S_3	S_3, 0	S_3, 0

下面在定义 1 的基础上给出如下定义:

定义 2　一个变量有限状态机(VFSM) M_v 用一个七元组表示如下:

$M_v = (I, O, S, \delta, \lambda, V, \zeta)$

其中:

I, O, S 和定义 1 中的含义相同;

V: 表示一组变量的集合 $V = \{V_1, V_2, \cdots, V_n\}$, n 表示M_v 中变量的个数. 其中变量 V_i 又表示一组指定值的集合.

δ: $D_T \rightarrow S$ 的状态转换函数, 其中 $D_T \subseteq D$, $D = S \times I \times V_1 \times V_2 \times \cdots \times V_n$;

λ: $D_T \rightarrow O$ 的输出函数;

ζ: 是变量转换函数的结合, 判断当一个转换执行时, 任何一个变量的值是否被修改.

图 2 是一个简单的用户界面模型, 在该界面中, 包括三个屏幕界面, S_0 表示主菜单界面, 用户可以从该界面选择进入数据输入界面 S_1 和保存界面 S_2. 在数据输入界面 S_1, 用户可以反复输入数据或返回 S_0. 在保存界面 S_2, 用户可以选择保存数据返回 S_0

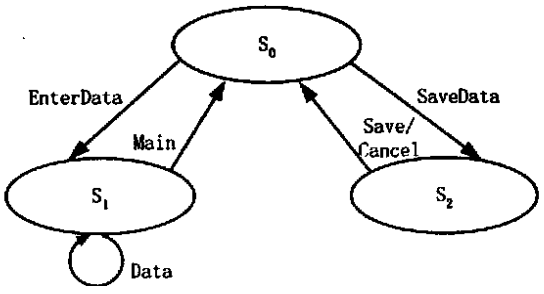

图 2　一个简单的界面模型

Fig. 2　A simple model of interface

或选择取消返回 S_0. 进入界面保存 S_2 只有在数据输入界面已输入数据后才有效. 该界面模型的 FSM 和 VFSM 界面模型分别如图 3 和图 4 所示.

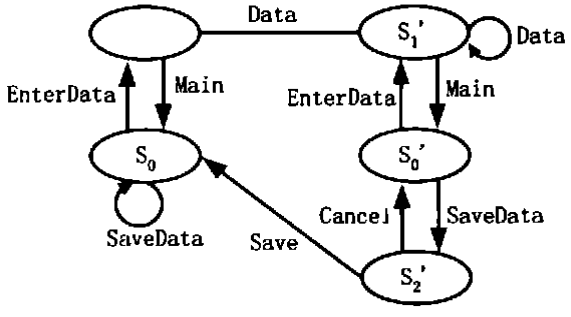

图 3　FSM 界面模型

Fig. 3　FSM model of interface

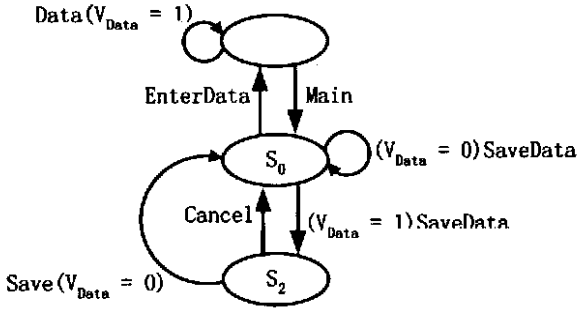

图 4　VFSM 界面模型

Fig. 4　VFSM model of interface

在上述两个界面模型中, FSM 界面模型使用了 5 个状态, VFSM 模型使用了 3 个状态. 在后者中, 引入了一个变量 $V_{Data} = \{0, 1\}$, 有些状态转换依赖变量 V_{Data} 之值. 图中在输入左边的变量表达式表示状态转换时给变量的必须满足的值, 而在输入右边的变量表达式表示状态转换后变量需修改的值.

2　VFSM $\rightarrow$ FSM 的转化定理和转化算法

本节首先给出一个 VFSM $\rightarrow$ FSM 的转化定理并

进行了证明,然后根据该转化定理给出了相应的转化算法.

在给出定理 1 之前,先给出如下定义:

定义 3 状态集 $S_{eq} = \{S_i \mid S_i \in S \times V_1 \times V_2 \times \cdots \times V_n\}$ 是 *VFSM* 的一个等价状态集.

定理 1 任何一个 VFSM 都至少可以等价一个 FSM.

证明:设 M_v 是一个 VFSM,构造一个新的转换函数:

$$\delta_q : S_{eq} \times I \rightarrow S_{eq}$$

δ_{eq} 是 δ 函数和 ζ 函数的组合函数. 这是可能的,因为 δ 函数和 ζ 函数有相同的输入域,并且 δ 函数的值域为 S,而 ζ 函数的值域为 $V = \{V_1, V_2, \cdots, V_n\}$.

因此,S_{eq} 是 δ 函数和 ζ 函数的值域的笛卡尔乘积.

同理,构造一个输出函数: $\lambda_{eq} : S_{eq} \times I \rightarrow O$.

存在一个 FSM $M = \{S_{eq}, I, O, \delta_{eq}, \lambda_{eq}\}$,定理得证.

上述证明过程给出了一种 VFSM 转换成等价的 FSM 的方法,该方法的缺点在于在集合 S_{eq} 中存在一定数量的不必要的附加状态. 下面给出一个 VFSM → FSM 并消除额外状态的伪代码算法.

算法 1 一个 VFSM → FSM 并消除额外状态的算法

定义一个访问状态队列 Queue 和 S_0 可到达的状态的状态列表 List;

```
//根据 VFSM 构造 Seq, δeq, λeq
construct Seq from VFSM;
construct Teq from VFSM;
construct Feq from VFSM;
//初始化状态队列
Queue. Insert (S0);
//初始化 S0 可到达的状态的状态列表
List. Insert (S0);
//构造 S0 可到达的状态的状态列表
while (Queue is not empty)
{
   从 Queue 取出状态队列中的当前状态 s;
   for(对于状态 s 中的每一个转换 t)
   {
      d 表示状态 s 的执行 t 转换的目标状态;
      if(d 不在 List 中)
      {
         List. Insert(d);
         Queue. Insert(d);
      }
   }

}
//删除额外的状态
for(对 Seq 中的每一个状态 s)
{
   if(s 不在 List 中)
   {
      delete(s);
   }
}
```

3 基于 VFSM 的 GUI 测试方法的步骤

该方法分为如下 5 个不同的步骤:VFSM 模型的定义和构造;VFSM → FSM 的转化;测试的生成;测试的执行和测试结果的分析. 具体说明如下:

1) 模型定义:通过一个列表来描述界面规范中所有状态[STATES]、输入[INPUTS]、输出[OUTPUTS]、状态转换[TRANSITIONS]、进行状态转换时所要求的变量值以及状态转换需要修改的变量值等信息来构造一个界面的 VFSM 模型.

下面三句语法分别用来描述状态转换、状态转换所需要的变量值和状态转换后修改的变量值.

@ arc < State> < Input> < NextState> < Ouput>

@ req < Variable> < Value Required>

@ set < Variable> < New Value>

如图 2 所示界面的 VFSM 采用列表形式表 2 所示.

2) VFSM → FSM 的转化:将 VFSM 作为一个输入,自动地产生一个输出. 在理论上,算法 1 可以实现一个 VFSM 到 FSM 的转化,实际上,在具体的转化时还需要有一些额外的考虑,如对状态转换时的空输出问题,关于识别序列的长度问题,使用文献[1]中 W_p 方法和哈希输出技术可以实现 VFSM →

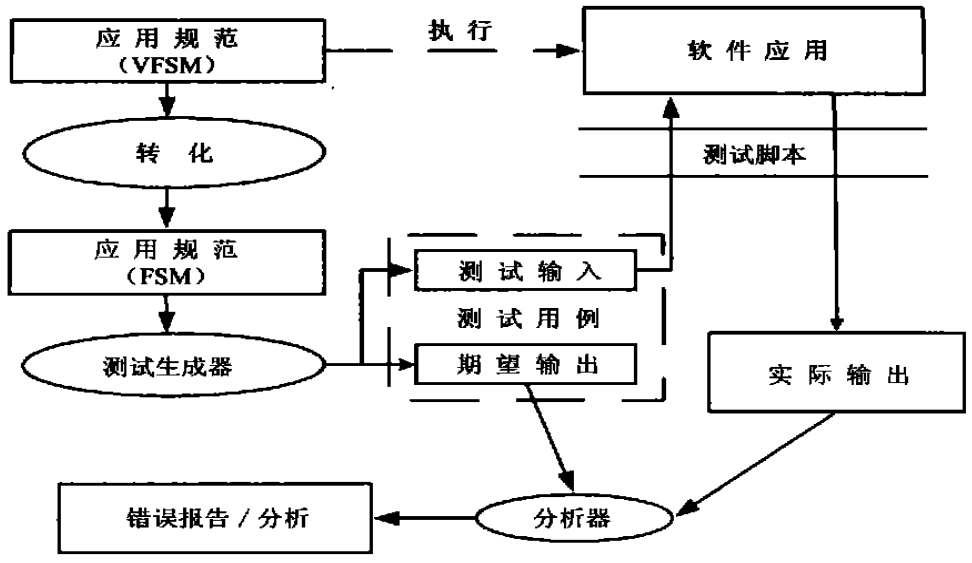

图 5　基于 VFSM 的GUI 测试的步骤

Fig. 5　The testing steps of GUI based on VFSM

表 2　一个简单的 VFSM

Tab. 2　A simple VFSM

[INPUTS]				
SavaData	Save	Data		
EnterData	Cancel	Main		
[OUTPUTS]				
O_0	O_1	O_2		
[VARIABLES]				
V_{Data}				
[STATES]				
S_0	S_1	S_2		
[TRANSITIONS]				
@ arc	S_0	EnterData	S_1	O_0
@ arc	S_0	SaveData	S_2	O_0
@ req	V_{data}	1		
@ arc	S_1	Main	S_0	O_1
@ arc	S_1	Data	S_1	O_1
@ set	V_{data}	1		
@ arc	S_2	Save	S_0	O_2
@ set	V_{data}	0		
@ arc	S_2	Cancel	S_0	O_2

FSM 的转化.

3) 测试的生成: 根据FSM 生成GUI 测试输入序列和相应的期望输出序列. 基于 FSM 的测试生成方法, 见文献[2].

4) 测试的执行: 将FSM 产生的输入序列应用被测的GUI 系统, 并产生一组实际的输出.

5) 测试结果分析: 比较由步骤(3) 生成的期望输出和步骤(4) 得出的实际输出, 从而检查 GUI 的错误, 并生成错误报告.

上述方法的执行步骤如图 5 所示.

4　应用实例和结论

采用该方法对笔者开发的一个 GUI 系统 SEAS1.0 进行了测试, 得到 VFSM 和 FSM 模型的测试数据如下表所示:

表 3　VFSM 和 FSM 测试数据的比较

Tab. 3　Comparison of testing data between VFSM and FSM

Type	States	Output	Transitions
VFSM	20	20	93
FSM	580	20	26 680

上述数据表明, 采用 VFSM 来模拟 GUI 的状态和状态转换数量大大地减少. 但该方法在构造界面的 VFSM 时, 仍然具有很大的难度, 必须依靠辅助工具从GUI 设计规范中生成 GUI 的VFSM 信息. 另外, 根据FSM 生成测试序列数量很大, 必须采用有效的方法对测试序列进行简化, 才能达到对 GUI 的有效测试.

参考文献:

[1] Fujiwara S, Bochmann G V, Khendek F, et al. Test selection based on finite state models[J]. IEEE Trans. on Soft Eng., 1991, 17(6): 591- 603.

[2] Lee D, Yannakakis M. Principles and methods of testing finite state machines — A survey[J]. Proc. of the IEEE, 1996, 84(8): 1 090- 1 123.

A Method of Testing GUI by Using VFSM

CHEN Qi-an[1], LI Xiao-jiang[2], LI Shu-yu[2]

(1. Dept. of Computer Science, Xiamen University, Xiamen 361005, China;
2. Dept. of Computer Science & Engineering, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: While FSMs do not make particularly good models for modern graphic user interface. We propose a new method of testing GUI by using VFSMs in this paper. We give the definition of VFSMs and a algorithm for conversion of a VFSM to FSM, and then, discuss the steps of this method. This method was applied to testing a GUI system, it shows that the states and transitions are obviously reduced.

Key words: FSM; VFSM; GUI; testing GUI; test; software testing; user interface