

DOI: 10.7652/xjtuxb201603020

# 业务流程建模与测试方法研究

丁明<sup>1,2</sup>, 张书玲<sup>1</sup>, 张琛<sup>3</sup>

(1. 西北大学信息科学与技术学院, 710127, 西安; 2. 中航工业西安航空计算技术研究所, 710119, 西安;  
3. 西安电子科技大学计算机学院, 710071, 西安)

**摘要:** 针对复杂业务流程设计测试效率低、自动化程度不高、测试用例正确性难以保证的问题,在研究接口自动机模型的基础上,提出了一种基于模型的业务流程测试方法。该方法首先采用扩展带约束的接口自动机对业务流程设计进行形式化描述,并给出了从业务流程设计模型到带约束的接口自动机模型的转换算法;然后基于模型完成了对业务流程设计与需求的一致性验证,将验证后的接口自动机模型作为业务流程的测试模型,通过其特有的“乐观方法”和博弈思想实现了流程间的嵌套调用组合;最后定义了业务流程的测试覆盖准则,并在满足活动约束条件组合覆盖准则的前提下,设计了相应的测试用例生成算法。实例分析表明:该方法能够简化测试过程,可用于业务流程设计的测试用例自动化生成,有效保证了测试用例的正确性,提高了测试效率。

**关键词:** 业务流程;接口自动机;覆盖准则;测试用例

**中图分类号:** TP311 **文献标志码:** A **文章编号:** 0253-987X(2016)03-0127-06

## Modeling and Testing Methods of Business Process

DING Ming<sup>1,2</sup>, ZHANG Shuling<sup>1</sup>, ZHANG Chen<sup>3</sup>

(1. School of Information Science and Technology, Northwest University, Xi'an 710127, China;  
2. Xi'an Aeronautics Computing Technique Research Institute, Aviation Industry Corporation of China, Xi'an 710119, China;  
3. School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

**Abstract:** To improve the automation level as well as the efficiency and accuracy of complex business process, and on the basis of studying interface automata models, a novel model-based testing approach for business processes is proposed. First, the extended constraint interface automata are used to describe the business process design. The algorithm for transforming the design models into constraint interface automata is presented. Furthermore, the consistency of business process design with requirements is verified. The verified interface automata are regarded as the test models of business processes. Optimistic approach and game theory are used to model the nested calls among business processes. Finally, the coverage criteria of business process test are defined. And the algorithm for test cases generating is given to satisfy the constraint combinations. Case studies show that this approach can be used for automatic generation of test cases for business process design and reduce the complexity of test cases generation and hence ensuring the correctness of test cases.

**Keywords:** business process; interface automata; coverage criteria; test case

复杂业务流程节点数量多、活动及关系复杂,且 流程之间相互嵌套调用,从而导致测试复杂度较高。

通过对业务流程的节点、活动、约束、迁移关系等构成元素进行抽象,并采用形式化方法进行描述,可实现对业务流程设计与需求的一致性以及流程的死锁、可达性等验证<sup>[1-4]</sup>。通过验证的形式化模型可作为业务流程的测试模型,依据测试覆盖准则,实现基于模型的测试用例自动化生成<sup>[5]</sup>,能够有效保证测试用例的正确性,提高业务流程设计的测试效率。

对业务流程的验证和测试,目前已提出了多种形式化模型。文献[6]中应用有限状态机对业务流程进行建模和一致性验证,并根据迁移覆盖和状态覆盖生成测试用例;文献[7]使用 Petri 网对业务流程进行建模, Petri 网具有直观的图形化表示、精确的语义、强大的表达能力和基于状态而不是基于事件等优点;文献[8]应用 Pi 演算进行 Web 服务组合验证,作为一种理论成熟的进程代数方法, Pi 演算使用基于文本的进程表达式描述系统,具有形式简单、表达能力强等优点。接口自动机 (Interface Automata) 是一种轻量级的形式化模型<sup>[9]</sup>,采用“乐观方法”来定义组合问题,并用博弈理论进行解释,本文采用带约束的接口自动机 (CIA) 作为描述业务流程的形式化模型。

## 1 业务流程建模

### 1.1 业务流程的定义与表示

Hammer 等给出了业务流程的经典定义:业务流程由一组创造价值的活动组合而成,活动有且仅有一个输入,不少于一个输出结果,输出结果表示客户价值的增值<sup>[10]</sup>。典型的业务流程包括输入资源、按一定规则执行的活动、活动之间的相互关系和作用(即结构)、输出结果、顾客、流程创造的价值等要素。为了准确、完整地描述流程执行过程,采用十二元组定义  $M = (O, A_i, A_e, A_o, A, S, E, C, \beta, \phi, \alpha, t)$ , 其中:  $O$  是流程中所有节点的集合;  $A_i$  是节点中的内部活动,其执行与其他流程无交互关系,活动执行后转至其他节点;  $A_e$  是节点中的输入活动,必须等待其他流程的输入激活,活动执行后转至其他节点;  $A_o$  是节点中的输出活动,其执行必定激活其他流程,同时流程转至其他节点;  $A$  是流程中所有活动的集合,  $A = A_i \cup A_e \cup A_o$ ;  $S$  是开始节点,每个流程有且只有唯一的开始节点;  $E$  是终止节点的集合,每个流程至少有一个终止节点;  $C$  是活动约束条件的集合;  $\beta$  是  $A_i$ 、 $C$  到  $O$  的一个函数关系,  $\beta(c_o, a_i) \in O, c_o \in C, a_i \in A_i$  是  $a_i$  所属节点和执行的约束条件;  $\phi$  是  $A_e$ 、 $C$  到  $O$  的一个函数关系,  $\phi(c_o, a_e) \in O, c_o \in C, a_e$

$\in A_e$  是  $a_e$  所属节点和执行的约束条件;  $\alpha$  是  $A_o$ 、 $C$  到  $O$  的一个函数关系,  $\alpha(c_o, a_o) \in O, c_o \in C, a_o \in A_o$  是  $a_o$  所属节点和执行的约束条件;  $t$  是流程节点之间转移的迁移函数,  $O \times A \rightarrow O, t(o, a) = o', o \in O, o' \in O, a \in A$  表示从一个节点执行活动后到达另一个节点。

### 1.2 业务流程的扩展接口自动机模型

文献[9]中 Luca 等提出了接口自动机理论,可用于描述组件接口的时序性,即接口交互活动间的先后顺序关系。接口自动机采用“乐观方法”判别接口组合兼容性,可大幅削减组合后的接口状态,是一种轻量级的自动机模型。

为了准确地描述复杂业务流程中活动执行的约束条件,本文将接口自动机扩展为 CIA,作为描述业务流程的形式化模型。定义 CIA 的十四元组  $P = (V_P, V_P^{\text{init}}, V_P^{\text{end}}, A_P, A_P^i, A_P^o, A_P^H, T_P, C_P, M_P, M_P^i, M_P^o, M_P^H, K_P)$ , 其中:  $V_P$  是自动机中所有状态的集合;  $V_P^{\text{init}} \subseteq V_P$  是初始状态的集合,集合中有且仅有一个状态;  $V_P^{\text{end}} \subseteq V_P$  是终止状态集,  $V_P^{\text{end}}$  至少包含一个状态;  $A_P$  为所有活动的集合,  $A_P = A_P^i \cup A_P^o \cup A_P^H$ ,  $A_P^i$  为输入活动集合,  $A_P^o$  为输出活动集合,  $A_P^H$  为内部活动集合,且它们之间互不相交;  $T_P \subseteq V_P \times A_P \times V_P$  是迁移的集合,将  $(v, a, v') \in T_P$  记为  $v \xrightarrow{a} pv'$ , 迁移的活动表示为  $\text{label}(v \xrightarrow{a} pv') = a$ , 开始节点  $\text{head}(v \xrightarrow{a} pv') = v$ , 结束节点  $\text{tail}(v \xrightarrow{a} pv') = v'$ ;  $C_P$  是所有输入活动、输出活动和内部活动的约束集合,若  $C_P = \emptyset$ ,则表示所有活动的执行均无约束;  $M_P^i \subseteq C_P \times A_P^i, M_P^o \subseteq C_P \times A_P^o, M_P^H \subseteq C_P \times A_P^H$  分别是输入活动、输出活动和内部活动与约束条件结合的集合;  $M_P = M_P^i \cup M_P^o \cup M_P^H$  是所有活动和约束结合的集合,记为  $a[c]$ ,  $(c, a) \in M_P, a \in A_P, c \in C_P$ ;  $K_P \subseteq M_P \times T_P$  是带约束的活动迁移的集合,将  $(v, c, a, v') \in K_P$  记为  $v \xrightarrow{a[c]} pv'$ , 集合中的迁移活动  $\text{label}(v \xrightarrow{a[c]} pv') = a$ , 开始节点  $\text{head}(v \xrightarrow{a[c]} pv') = v$ , 结束节点  $\text{tail}(v \xrightarrow{a[c]} pv') = v'$ , 迁移活动的约束条件  $\text{cons}(v \xrightarrow{a[c]} pv') = c$ 。

定义带约束的接口自动机  $P$  的一个执行片段是状态与活动(含约束)交替排列的有限序列  $v_0 a_0 [c_0] v_1 a_1 [c_1] \cdots v_n$ , 其中  $(v_i, c_i, a_i, v_{i+1}) \in K_P, 0 \leq i < n$ 。任给两个状态  $u, v \in V_P$ , 若存在一个执行片段, 第一个状态为  $u$  且最后一个状态为  $v$ , 称  $v$  从  $u$  带

约束可达。

为了实现业务流程的 CIA 描述,定义复杂业务流程的带约束的接口自动机模型构造算法。

### 算法 1 构造业务流程带约束的接口自动机模型

**输入** 业务流程的十二元组描述

**输出** 业务流程的带约束的接口自动机模型

(1) CIA 的状态集合  $V_P = O$ , 业务流程中的一个节点对应一个状态;

(2) 创建初始状态  $V_P^{\text{init}} = S$ , 业务流程的开始节点对应 CIA 的初始状态;

(3) 创建终止状态集合  $V_P^{\text{end}} = E$ , 业务流程的结束节点对应终止状态;

(4) 内部活动集合  $A_P^{\text{H}} = A_i$ , 输入活动集合  $A_P^{\text{I}} = A_e$ , 输出活动集合  $A_P^{\text{O}} = A_o$ ;

(5) 所有活动集合  $A_P = A_P^{\text{I}} \cup A_P^{\text{O}} \cup A_P^{\text{H}}$ ;

(6) 状态迁移集合, 包含迁移前后状态和活动之间的关系,  $T_P = \{(v, a, v') \mid v \in O \wedge v' \in O \wedge a \in A \wedge t(v, a) = v'\}$ ;

(7) 约束条件集合  $C_P = C$ , 业务流程活动约束条件对应自动机的活动约束;

(8) CIA 输入活动、输出活动和内部活动与约束条件的结合集合分别为  $M_P^{\text{I}} = \{(c, a) \mid (c, a) \in \varphi\}$ ,  $M_P^{\text{O}} = \{(c, a) \mid (c, a) \in \alpha\}$ ,  $M_P^{\text{H}} = \{(c, a) \mid (c, a) \in \beta\}$ ;

(9) CIA 的所有活动与其约束条件的结合集合  $M_P = M_P^{\text{I}} \cup M_P^{\text{O}} \cup M_P^{\text{H}}$ ;

(10) CIA 的带约束的活动迁移集合  $K_P \subseteq M_P \times T_P = \{(v, c, a, v') \mid (c, a) \in M_P \wedge (v, a, v') \in T_P\}$ 。

### 1.3 业务流程间嵌套调用建模

业务流程之间相互嵌套调用时, 执行业务流程中的输出活动, 会向外部发送调用消息, 启动另一业务流程; 另一方面, 输入活动激活需等待另一流程发送的消息。多个业务流程互相交互调用的过程, 在设计测试用例时可进行组合, 成为一个组合的业务流程, 从而简化测试用例生成。业务流程的相互调用可转化为带约束的接口自动机组合, 两个带约束的接口自动机  $P$  和  $Q$ , 组合的过程需先判断  $P$  和  $Q$  中带约束的活动是否满足组合条件。

**定义** 两个接口自动机  $P, Q$  若满足条件

$$\begin{aligned} A_P^{\text{H}} \cap A_Q &= \emptyset, A_P^{\text{I}} \cap A_Q^{\text{I}} = \emptyset, \\ A_Q^{\text{H}} \cap A_P &= \emptyset, A_P^{\text{O}} \cap A_Q^{\text{O}} = \emptyset, \\ (M_P^{\text{I}} \cap M_Q^{\text{O}}) \cup (M_P^{\text{O}} \cap M_Q^{\text{I}}) &\neq \emptyset \end{aligned}$$

则称它们是可组合的。

由定义知,  $P, Q$  不能有相同的输入或输出活动, 一方的内部活动不能与另一方的任何活动相同,

存在一方带约束的输入活动与另一方带约束的输出活动相同。满足条件的带约束的接口自动机  $P, Q$  是可组合的, 记为  $P \times Q$ , 可表示为  $P \times Q = (V_{P \times Q}, V_{P \times Q}^{\text{init}}, V_{P \times Q}^{\text{end}}, A_{P \times Q}, A_{P \times Q}^{\text{I}}, A_{P \times Q}^{\text{O}}, A_{P \times Q}^{\text{H}}, T_{P \times Q}, C_{P \times Q}, M_{P \times Q}, M_{P \times Q}^{\text{I}}, M_{P \times Q}^{\text{O}}, M_{P \times Q}^{\text{H}}, K_{P \times Q})$ 。为更加清晰的说明, 定义  $\text{share}(P, Q) = (M_P^{\text{I}} \cap M_Q^{\text{O}}) \cup (M_P^{\text{O}} \cap M_Q^{\text{I}})$ , 表示  $P, Q$  一方带约束的输入活动与另一方带约束的输出活动相同的活动集合。 $P \times Q$  的集合定义为  $V_{P \times Q} = V_P \times V_Q$ ;  $V_{P \times Q}^{\text{init}} = V_P^{\text{init}} \times V_Q^{\text{init}}$ ;  $V_{P \times Q}^{\text{end}} = V_P^{\text{end}} \times V_Q^{\text{end}}$ ;  
 $A_{P \times Q}^{\text{I}} = \{a \mid (a \in A_P^{\text{I}} \cup A_Q^{\text{I}}) \wedge a \notin \text{share}(P, Q)\}$ ;  
 $A_{P \times Q}^{\text{O}} = \{a \mid (a \in A_P^{\text{O}} \cup A_Q^{\text{O}}) \wedge a \notin \text{share}(P, Q)\}$ ;  
 $A_{P \times Q}^{\text{H}} = \{a \mid (a \in A_P^{\text{H}} \cup A_Q^{\text{H}}) \vee a \in \text{share}(P, Q)\}$ ;  
 $A_{P \times Q} \subseteq A_{P \times Q}^{\text{I}} \cup A_{P \times Q}^{\text{O}} \cup A_{P \times Q}^{\text{H}}$ ;  $C_{P \times Q} = C_P \cup C_Q$ ;  
 $A_{P \times Q} = A_{P \times Q}^{\text{I}} \cup A_{P \times Q}^{\text{O}} \cup A_{P \times Q}^{\text{H}}$ ;  
 $M_{P \times Q}^{\text{I}} \subseteq C_{P \times Q} \times A_{P \times Q}^{\text{I}}$ ;  $M_{P \times Q}^{\text{O}} \subseteq C_{P \times Q} \times A_{P \times Q}^{\text{O}}$ ;  
 $M_{P \times Q}^{\text{H}} \subseteq C_{P \times Q} \times A_{P \times Q}^{\text{H}}$ ;  
 $M_{P \times Q} \subseteq M_{P \times Q}^{\text{I}} \cup M_{P \times Q}^{\text{O}} \cup M_{P \times Q}^{\text{H}}$ ;  $T_{P \times Q} = \{((v, u), a, (v', u)) \mid (v, a, v') \in T_P \wedge a \notin \text{share}(P, Q) \wedge u \in V_Q\} \cup \{((v, u), a, (v, u')) \mid (u, a, u') \in T_Q \wedge a \notin \text{share}(P, Q) \wedge v \in V_P\} \cup \{((v, u), a, (v', u')) \mid (v, a, v') \in T_P \wedge (u, a, u') \in T_Q \wedge a \in \text{share}(P, Q)\}$ ;  
 $K_{P \times Q} \subseteq M_{P \times Q} \times T_{P \times Q}$ 。

两个 CIA 组合后形成的  $P \times Q$  中可能存在某个状态下  $P$  中带约束的输出活动不能满足  $Q$  中带约束的输入活动, 或在某个状态下  $Q$  中带约束的输出活动不能满足  $P$  中带约束的输入活动, 即两个待组合的 CIA 中带约束的输入和输出活动不匹配, 定义为到达了非法状态, 组合是不兼容的。

通过上述方法, 可实现业务流程的形式化描述, 可转化为带约束的接口自动机, 通过模型检测方法验证业务流程设计模型与需求的一致性, 以及流程的死锁、可达性等。经过验证的业务流程 CIA 模型将用于测试用例的生成。

## 2 基于接口自动机的测试用例生成

本文定义的复杂业务流程的测试用例生成过程如图 1 所示, 使用 CIA 模型形式化描述业务流程, 并进行流程设计与需求的一致性验证, 确认后的 CIA 模型可采用“乐观方法”处理组合问题, 描述流程间的嵌套调用。针对合并后的 CIA 模型, 定义覆盖准则, 设计算法, 自动生成满足覆盖准则的测试用例。

### 2.1 覆盖准则定义

为了使设计的测试用例满足要求的覆盖程度,

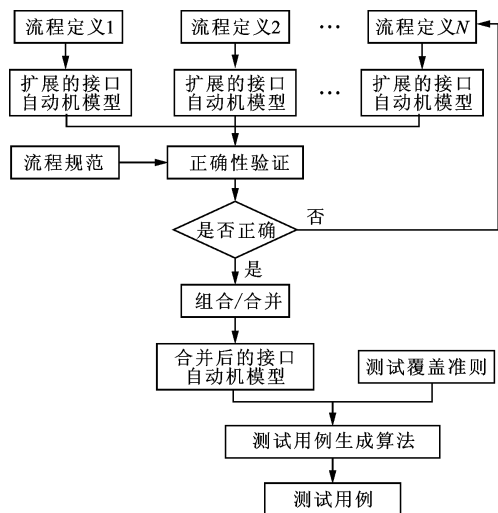


图 1 复杂业务流程的测试用例生成过程

测试用例生成需遵循一定的覆盖标准。定义一个测试集合  $D$  和一个待测试的业务流程  $M$ ，测试的覆盖标准包括：①节点覆盖， $D$  必须保证  $M$  中的每个节点至少运行一次，节点覆盖对活动覆盖较少，是较容易满足的测试准则，需要的测试用例相对较少；②活动覆盖， $D$  必须保证  $M$  中的每个活动至少执行一次，活动覆盖比节点覆盖强，但覆盖程度仍不高；③活动约束条件覆盖， $D$  不仅保证业务流程中每个活动至少执行一次，而且对活动约束条件表达式中每个条件变量各种可能结果都执行一次，通常情况下活动约束条件覆盖通常比活动覆盖强，因为活动覆盖只遍历整个活动约束条件表达式的值，而活动约束条件覆盖使每个条件变量都取得不同的结果；④活动约束条件组合覆盖， $D$  保证覆盖业务流程所有活动约束条件表达式中所有条件变量的各种组合可能都至少出现一次。

上述的 4 类覆盖标准中，活动约束条件组合覆盖是最强的覆盖准则，符合该准则的测试用例，可以满足活动覆盖和活动约束条件覆盖标准。本文在满足活动约束条件组合覆盖准则的前提下，给出业务流程的测试用例定义和生成过程。

## 2.2 测试用例生成

业务流程的测试用例(BPTC)是为了达到用户的业务需求目标而设计的一组测试输入、执行条件和预期结果，用以确认业务流程中的某个场景是否满足用户需求。BPTC 是流程中从开始节点到终止节点的一个执行序列，在使用 CIA 模型描述的业务流程设计中，包含了产生测试用例所需的信息，一个测试用例对应从初始状态到某个终止状态的带约束可达的执行片段。根据业务流程活动约束条件组合

覆盖准则和测试用例定义，测试用例设计与生成可采用以下算法。

### 算法 2 业务流程测试用例集生成算法

输入 带约束的接口自动机  $P$

输出 测试用例集合  $D$

(1) 遍历  $P$  中约束条件集合  $C_P$ ，列出所有约束条件。

(2) 遍历集合  $C_P$  中每个约束条件变量各种可能取值，生成约束条件取值的全排列组合。

(3) 经过验证的 CIA 模型应满足对任意约束条件组合均存在从初始状态  $v \in V_P^{\text{init}}$  到终止状态  $u \in V_P^{\text{end}}$  的带约束可达的执行片段。因此，按照每个组合序列，从  $P$  的初始状态  $v \in V_P^{\text{init}}$  开始，根据约束条件进行状态遍历，到终止状态  $u \in V_P^{\text{end}}$  结束，在状态遍历中循环限制至多出现一次。

(4) 通过以上的步骤，在  $P$  中为每个约束条件组合序列对应生成一个从初始状态到终止状态的执行片段。一个执行片段对应生成一条测试用例，所有执行片段集合即为  $D$ 。

## 3 实例分析

本节以简化的银行理财产品购买流程为例，说明业务流程的建模与测试用例生成过程。根据产品风险特性，一般将理财产品风险由低到高分 5 个级别，用户允许购买的理财产品风险必须小于等于用户可接受的风险等级。用户购买理财产品的业务流程包括登录系统、确认用户类型、选择理财产品、输入购买金额、执行购买等过程，流程执行过程中嵌套调用用户可接受风险测评和计算理财产品余额两个子流程。使用业务流程建模与标注(BPMN)标准规范描述的流程如图 2、3、4 所示。

根据定义，图 2、3、4 所示业务流程分别表示为  $M_1$ 、 $M_2$ 、 $M_3$  3 个十二元组。使用算法 1 定义的构造方法，流程对应的 CIA 模型  $P_1$ 、 $P_2$ 、 $P_3$ ，如图 5、6、7 所示。模型中输入活动用“?”表示，输出活动用“!”表示，未标识的为内部活动。

$P_1$ 、 $P_2$ 、 $P_3$  满足定义的模型组合条件，组合后的 CIA 模型表示为  $Q=(P_1 \times P_2) \times P_3$ ，如图 8 所示。

根据算法 2 定义的测试用例集生成方法，组合后的接口自动机模型  $Q$  中所有约束条件取值的组合如表 1 所示，测试用例集合为  $D$ 。按照每个组合序列，从 CIA 模型的初始状态开始，根据约束条件取值进行状态遍历到终止状态结束，状态遍历中循环限制至多出现一次，得到测试用例集合  $D$  包含 6

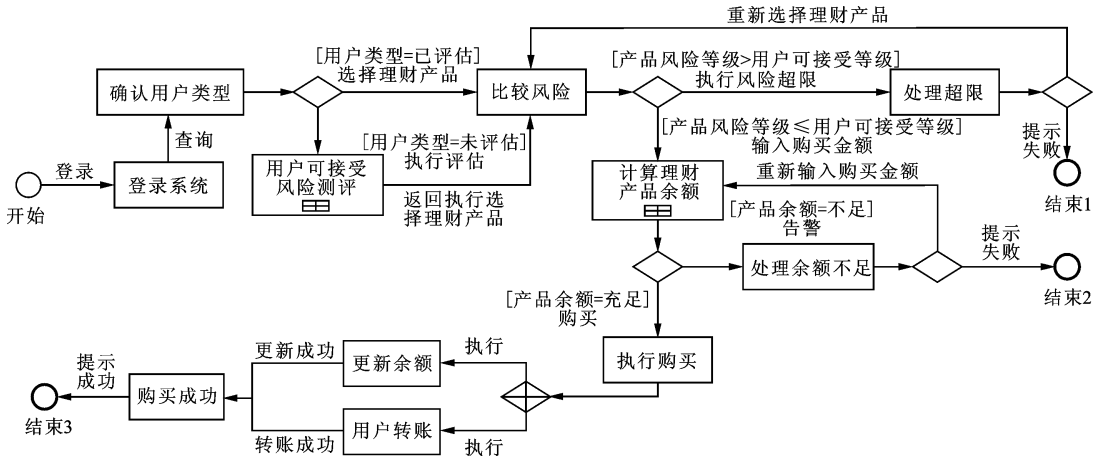


图 2 理财购买流程

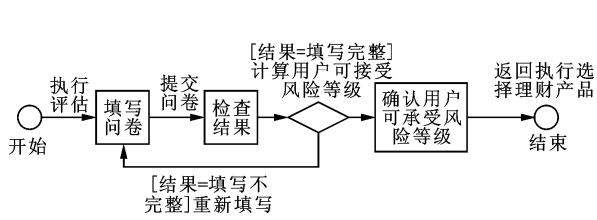


图 3 用户可接受风险测评流程

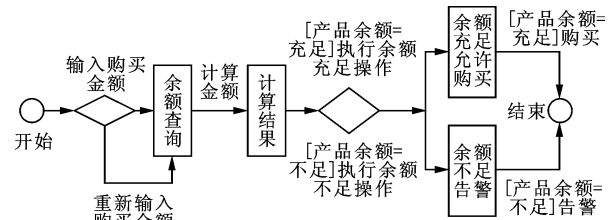


图 4 计算理财产品余额流程

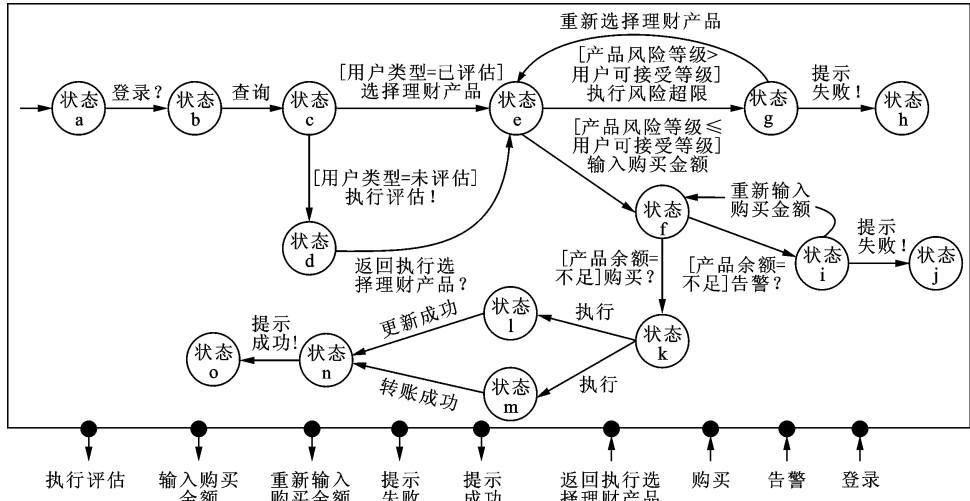


图 5  $M_1$  流程的 CIA 模型  $P_1$

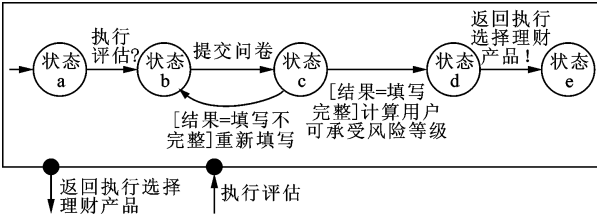


图 6  $M_2$  流程的 CIA 模型  $P_2$

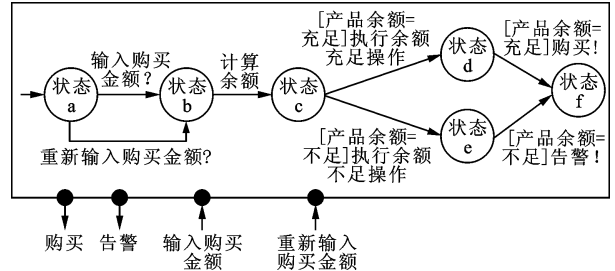


图 7  $M_3$  流程的 CIA 模型  $P_3$

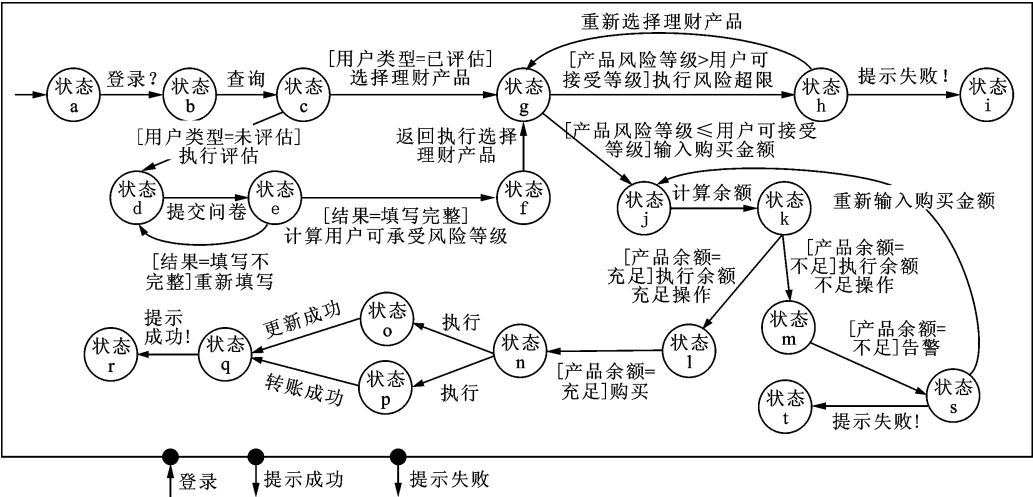


图 8 组合后的 CIA 模型 Q

条测试用例,可以覆盖表 1 约束条件取值组合,每条测试用例对应带约束的接口自动机 Q 中一个初始状态{状态 a}到终止状态集{状态 i,状态 t,状态 r}的执行片段。

表 1 约束条件取值组合

| 用户类型 | 产品风险       | 产品余额 |
|------|------------|------|
| 已评估  | >用户可接受风险等级 | 充足   |
| 已评估  | >用户可接受风险等级 | 不足   |
| 已评估  | ≤用户可接受风险等级 | 充足   |
| 已评估  | ≤用户可接受风险等级 | 不足   |
| 未评估  | >用户可接受风险等级 | 充足   |
| 未评估  | >用户可接受风险等级 | 不足   |
| 未评估  | ≤用户可接受风险等级 | 充足   |
| 未评估  | ≤用户可接受风险等级 | 不足   |

4 结束语

良好的业务流程设计与实现是保证企业流程运行的关键,对复杂的业务流程进行验证、测试是一项具有挑战性的工作,尤其是复杂流程间的相互嵌套调用。文中采用扩展的带约束条件的接口自动机描述业务流程设计模型,通过接口自动机特有的“乐观方法”和博弈思想处理流程间的嵌套调用组合,简化了测试过程,并定义了测试覆盖准则,实现测试用例的自动生成。将其应用于业务流程管理实施中,可实现测试工作的早期规划,提高测试的自动化程度和测试效率。

参考文献:

[1] 雷丽晖,段振华. 一种基于扩展有限自动机验证组合

Web 服务的方法 [J]. 软件学报, 2007, 18(12): 2980-2990.

LEI Lihui, DUAN Zhenhua. An extended deterministic finite automata based method for the verification of composite web services [J]. Journal of Software, 2007, 18(12): 2980-2990.

[2] 张广泉,狄浩军,石慧娟,等. 基于扩展自动机的服务组合静态与动态验证方法 [J]. 通信学报, 2012, 33(s1): 1-8.

ZHANG Guangquan, DI Haojun, SHI Huijuan, et al. Static and dynamic verifying method based on extended finite automata for service compositions [J]. Journal on Communications, 2012, 33(s1): 1-8.

[3] 骆翔宇,轩爱成,沙宗鲁. 基于时间自动机的 Web 服务模型检测 [J]. 计算机科学, 2010, 37(8): 139-144.

LUO Xiangyu, XUAN Aicheng, SHA Zonglu. Model checking web services based on timed automata [J]. Computer Science, 2010, 37(8): 139-144.

[4] CAMBRONERO M E, DÍAZ G, VALERO V, et al. Validation and verification of Web services choreographies by using timed automata [J]. The Journal of Logic and Algebraic Programming, 2011, 80(1): 25-49.

[5] LIU S, NAKAJIMA S. A compositional approach to automatic test case generation based on formal specifications [C]// Proceedings of the 4th International Conference on Secure Software Integration and Reliability Improvement. Piscataway, NJ, USA: IEEE, 2010: 147-155.

[6] DE MASELLIS R, MAGGI F M, MONTALI M. Monitoring data-aware business constraints with finite state automata [C]// Proceedings of International Conference on Software and System. New York, USA:

ACM, 2014: 134-143.

- [7] CLEMPNER J. Verifying soundness of business processes: a decision process Petri nets approach [J]. *Expert Systems with Applications*, 2014, 41 (11): 5030-5040.
- [8] YUAN M, HUANG Z, LI X, et al. Towards a formal verification approach for business process coordination [C]// *Proceedings of 2010 IEEE 8th International Conference on Web Services*. Piscataway, NJ, USA: IEEE, 2010: 362-368.
- [9] DE ALFARO L, HENZINGER T A. Interface autom-

ata [C]// *Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on the Foundations of Software Engineering*. New York, USA: ACM, 2001: 109-120.

- [10] HAMMER M, CHAMPY J. Reengineering the corporation: a manifesto for business revolution (Collins business essentials) [M]. New York, USA: HarperCollins, 2009.

(编辑 赵炜)

(上接第 119 页)

- [7] TANG J, LIU F, MENG Q H, et al. Partial discharge recognition through an analysis of SF<sub>6</sub> decomposition products: part 2 Feature extraction and decision tree-based pattern recognition [J]. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2012, 19(1): 37-44
- [8] 电力工业部. DL/T595—1996 六氟化硫电气设备气体监督细则 [S]. 北京: 中国电力出版社, 1996.
- [9] 国家标准化管理委员会. GB/T8905—2012 六氟化硫电气设备中气体管理和检测导则 [S]. 北京: 中国标准出版社, 2012.
- [10] 罗飞. 六氟化硫开关微水含量在线监测系统研究 [D]. 重庆: 重庆大学, 2004: 4-5.
- [11] BRUNT R J V. Production rates for oxyfluorides SOF<sub>2</sub>, SO<sub>2</sub>F<sub>2</sub>, and SOF<sub>4</sub> in SF<sub>6</sub> corona discharge [EB/OL]. [2015-06-16]. [www.researchgate.net/publication/272674029](http://www.researchgate.net/publication/272674029).
- [12] 唐炬, 王悠, 孙慧娟, 等. H<sub>2</sub>O 对局部放电下 SF<sub>6</sub> 分解生成特征组分的影响 [J]. *高电压技术*, 2014, 40 (3): 787-794.  
TANG Ju, WANG You, SUN Huijuan, et al. Influence of moisture on SF<sub>6</sub> decomposition characteristics induced by partial discharge [J]. *High Voltage Engineering*, 2014, 40(3): 787-794.
- [13] DING W D, ZHOU W W. Decomposition characteristics of SF<sub>6</sub> under partial discharges with point-to-plane electrode defect [J]. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2013, 20(5): 1727-1736.
- [14] 国家标准化管理委员会. GB/T12022—2006 工业六氟化硫 [S]. 北京: 中国标准出版社, 2006.
- [15] CHANTRY P J, WOOTTON R E. A critique of methods for calculating the dielectric strength of gas-

mixtures [J]. *Journal of Applied Physics*, 1981, 52 (4): 2731-2739.

- [16] VANBRUNT R J. Water vapor-enhanced electron-avalanche growth in SF<sub>6</sub> for nonuniform fields [J]. *Journal of Applied Physics*, 1986, 59(7): 2314-2323.
- [17] LEE L C, SMITH G P. Photodissociation and photo-detachment of molecular negative ions: VI ions in O<sub>2</sub>/CH<sub>4</sub>/H<sub>2</sub>O mixtures from 3 500 to 8 600 Å [J]. *The Journal of Chemical Physics*, 1979, 70 (4): 1727-1735.
- [18] VANBRUNT R J, HERRON J T. Plasma chemical model for decomposition of SF<sub>6</sub> in a negative glow corona discharge [J]. *Physica Scripta*, 1994, 53(6): 9-29.
- [19] SCHNEIDER H, WEBER J M. Infrared spectra of SF<sub>6</sub><sup>-</sup> • (H<sub>2</sub>O)<sub>n</sub> (n=1-3): incipient reaction and delayed onset of water network formation [J]. *Journal of Chemical Physics*, 2007, 127(10): 1-7.
- [20] 唐炬, 裴吟君, 曾福平, 等. 局部放电下微水对 SF<sub>6</sub> 分解组分的形成及其影响规律 [J]. *电工技术学报*, 2012, 27(10): 13-19.  
TANG Ju, QIU Yinjun, ZENG Fuping, et al. Formation mechanism and influence rules of trace levels H<sub>2</sub>O on SF<sub>6</sub> characteristic decomposition components under partial discharge [J]. *Transactions of China Electrotechnical Society*, 2012, 27(10): 13-19.
- [21] VANBRUNT R J, HERRON J T. Fundamental processes of SF<sub>6</sub> decomposition and oxidation in glow and corona discharges [J]. *IEEE Transactions on Electrical Insulation*, 1990, 25(1): 75-94.

(编辑 赵炜 杜秀杰)