

# 基于 UML 交互概览图的测试线索的生成方法

曾 一<sup>1,2</sup>, 王翠钦<sup>1\*</sup>, 李函逾<sup>1</sup>, 洪 豪<sup>1</sup>

(1. 重庆大学 计算机学院, 重庆 400030; 2. 软件理论与技术重庆市重点实验室, 重庆 400030)

(\* 通信作者电子邮箱 847870761@qq.com)

**摘要:**针对单一 UML 模型测试不充分的问题,结合 UML2.0 交互概览图的特点开展测试方法研究,提出一种测试线索自动生成的方法。首先,给出 UML 类图、顺序图、交互概览图(OID)的形式化定义;通过提取交互概览图的流程信息和对象交互信息分别构造节点控制流图(NCFG)和消息序列图(MSD);将从 MSD 提取的消息路径嵌入到 NCFG 中,构造可测试交互概览图模型;最后采用两两覆盖准则生成测试线索。实验验证了该方法自动生成的测试线索在保证测试充分性的前提下可避免组合爆炸。

**关键词:**UML 模型;交互概览图;流程信息;覆盖准则;测试线索

**中图分类号:**TP311.5 **文献标志码:**A

## Test clue generation based on UML interaction overview diagram

ZENG Yi<sup>1,2</sup>, WANG Cuiqin<sup>1\*</sup>, LI Hanyu<sup>1</sup>, HONG Hao<sup>1</sup>

(1. College of Computer Science, Chongqing University, Chongqing 400030, China;

2. Chongqing Key Laboratory of Software Theory and Technology, Chongqing 400030, China)

**Abstract:** Concerning the problem that single UML model can not test the software sufficiently, this paper proposed a new method of automatically generating software test clues by combining the characteristics of UML2.0 interaction overview diagram. First, this paper gave the formal definition of UML class diagrams, sequence diagrams and Interaction Overview Diagrams (IOD). Second, the Node Control Flow Graph (NCFG) was constructed by extracting the process information of the interaction overview diagram while the Message Sequence Diagrams (MSD) were constructed by extracting the object interaction information. The testable model of IOD was constructed by embedding the MSD's message path into NCFG. At last, the paper adopted two-two coverage criterion to generate test clues. The experiment verifies that this method which automatically generates test clues avoids the combinatorial explosion while guaranteeing the test adequacy.

**Key words:** UML model; Interaction Overview Diagram (IOD); process information; coverage criterion; test clue

## 0 引言

目前基于 UML 模型的软件测试是基于模型软件测试的主流选择。在基于 UML 模型软件测试<sup>[1-5]</sup>中,大多数是针对顺序图、状态图、活动图等单一模型进行测试研究的,而单一模型只能侧重反映系统的某个层面的信息,不能对系统进行充分和完备的测试。

交互概览图(Interaction Overview Diagram, IOD)<sup>[6-7]</sup>是对象管理组织(Object Management Group, OMG)在 UML2.0<sup>[8-10]</sup>中提出了一种新的图,能够在系统业务流程和对象间交互细节两个层面上反映系统的信息。交互概览图 IOD 有效地克服了 UML 活动图在对象间交互细节描述的欠缺以及 UML 顺序图不能有效地表达复杂对象交互及宏观更改的缺陷。

国外对交互概览图的形式化已有了一定的研究<sup>[6-7,11]</sup>,但没有针对交互概览图提取测试线索的研究,国内基于 UML2.0 交互概览图的软件测试较少。UML 交互概览图的特性决定了基于 UML 交互概览图的测试既可以获取较充分和完备的测试线索,测试线索是指从图的起始节点到结束节点的一条路径。

本文首先对 UML 交互概览图、类图、顺序图进行形式化定义,构造可测试交互概览图模型,针对构造的模型设计覆盖准则并生成测试线索。

## 1 UML 模型形式化定义

UML 模型的半形式化特性使得无法直接对 UML 模型进行信息的提取,因此需要对 UML 交互概览图及其相关的模型以数学的方式进行形式化描述。

### 1.1 UML 类图

类元组(Class Tuple, CT)<sup>[12]</sup>是面向对象中类的数学表示形式,类图中一个类就可以表示为一个类元组,类图表示为多个类元组的集合。类元组由类名、类属性以及类方法等基本信息组成。形式化定义如下:

$CT(C) := \langle \text{className}, \{ \langle \text{Attribute} \rangle \}, \{ \langle \text{Method} \rangle \} \rangle$

$\text{Attribute} := \langle \text{attrName}, \text{Type}, \text{Default Value} \rangle$

$\text{Method} := \langle \text{methName}, \{ \langle \text{Parameter} \rangle \}, \text{returnType} \rangle$

$\text{Parameter} := \langle \text{paraName}, \text{Type}, \text{Default Value} \rangle$

其中:当类 C 为空类时,用 NULL 来表示。

### 1.2 UML 顺序图

UML2.0 顺序图(Sequence Diagram, SD)在 UML1.x 顺序

收稿日期:2013-06-08;修回日期:2013-09-27。

**作者简介:**曾一(1961-),男,山西大同人,教授,CCF 会员,主要研究方向:软件工程、软件度量、软件测试、软件模型、软件工具、面向服务计算;王翠钦(1988-),女,河北沧州人,硕士研究生,主要研究方向:软件测试;李函逾(1990-),男,云南保山人,硕士研究生,主要研究方向:软件测试;洪豪(1988-),男,湖北崇阳人,硕士研究生,主要研究方向:面向服务计算。

图的基础上增加了组合框和交互引用的新特性。顺序图由顺序图名称、对象、消息和组合框等基本元素组成,定义如下五元组:SD := < Name, O, M, CF >。其中 M 代表消息集合,式(1)给出消息 Mi 的形式化定义;CF 为组合框的集合,式(3)给出了形式化定义。

$$M_i := \{ msgId, message, oSender, oReceiver, [/guard] \} \quad (1)$$

其中: message 为消息名称,对应类元祖中的方法 Method; guard 为所在组合框的监护条件,表示方法为“组合框名称@‘条件’”。

$$CF := \{ alt, opt, break, loop \} \quad (2)$$

其中:为了简化分析的复杂度,本文只考虑式(2)中给出的 4 种组合框的结构, break 不单独使用。

### 1.3 UML 交互概览图

交互概览图是一种特殊的活动图,活动节点是一个交互顺序图,活动边是控制流。将 UML2.0 交互概览图定义为一个六元组<sup>[6]</sup>: IOD := < n0, nf, B, D, I, E >。其中:n0 为 IOD 的初始节点; nf 为 IOD 的结束节点; B 代表分叉/汇合(fork/join)节点的集合; D 代表决策/合并(decision/merge)节点的集合; I 代表是交互(interaction)节点的集合,交互节点的名称即为顺序图片段(SD)的名称; E 是连接 IOD 节点的边的集合。

## 2 测试线索生成基本流程

基于 UML 交互概览图测试线索的流程如图 1 所示。

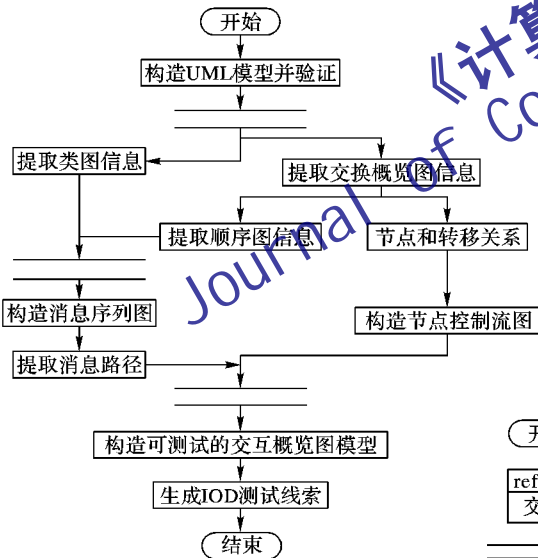


图 1 生成测试线索流程

- 1) 利用 UML 建模工具 RSA<sup>[13]</sup> 构建 UML 类图、交互概览图和顺序图片段,并验证模型一致性、正确性和完整性;
- 2) 提取模型信息并构造可测试交互概览图;
- 3) 设计覆盖准则并生成包含对象交互信息和系统流程信息的测试线索。

## 3 可测试交互概览图模型构造

### 3.1 节点控制流图构造

节点控制流图(Node Control Flow

Graph, NCFG)是对 UML 交互概览图的流程结构信息进行抽象转换的一种可测试的有向图,将交互顺序图片段抽象为一个活动节点,忽略其内部信息。根据 UML 交互概览图流程结构可将 NCFG 定义为四元组:NCFG := < n0, nf, N, E >,其中:n0 为 NCFG 的初始节点; nf 为 NCFG 的一个结束节点; N 为 NCFG 节点的集合,由式(3)~式(4)给出; E 是连接 NCFG 节点的边的集合,由式(5)给出。

$$N_i := \{ Name, Type \} \quad (3)$$

$$Type := \{ initial, fork, join, decision, merge, interaction, final \} \quad (4)$$

$$e_i := \{ N_i, N_{i+1} \} \quad (5)$$

IOD 流程结构到 NCFG 的转化规则为:

1) 节点的转换:将 IOD 中的各种类型节点转换为 NCFG 中的节点相应类型的节点 Ni; IOD 决策分支的条件抽象为 NCFG 的一个节点,如图 2 中的“条件 1”抽象为“N11”。

2) 边的转换:连接 IOD 节点的边的集合 E 对应连接 NCFG 节点的有向边的集合 E;为保证与 IOD 图的一致性,IOD 中决策节点 Ni 到 Ni+1 的边要转化为 NCFG 上添“Ni 到条件节点”“条件节点到 Ni+1”的两条边,如图 2 中 < N9, N11 >, < N11, N7 >。

按照转换规则将 IOD 图转换为 NCFG 如图 2 所示。

### 3.2 消息序列图构造

交互概览图的交互节点为顺序图片段,通过对顺序图中的消息和组合框的分析,将其转化为一种可测试的消息序列图(Message Sequence Diagram, MSD)。MSD 是由消息节点和节点间的边组成的有向图,将其定义为四元组:MSD := < V, E, Vs, Vf >。其中:V 是消息节点的集合,Vi 对应顺序图中的消息 Mi,式(6)中给出了 Vi 的定义;Vs 和 Vf 分别表示消息序列图 MSD 的开始节点和结束节点; E 是 MSD 边的集合,式(7)给出了 MSD 消息边的定义。

$$V_i := \langle M_i, [/guard], oSender, oReceiver \rangle \quad (6)$$

$$e_i := \{ \langle Pre(V_i), V_i \rangle | \langle V_i, Suc(V_i) \rangle \} \quad (7)$$

其中:Pre(Vi)是消息节点 Vi 的前驱, Suc(Vi)是消息节点 Vi 后继。

UML2.0 顺序图增加了组合框,为了区分独立消息和组合框,引入顺序图单元(Sequence Diagram Unit, SDU),为独立

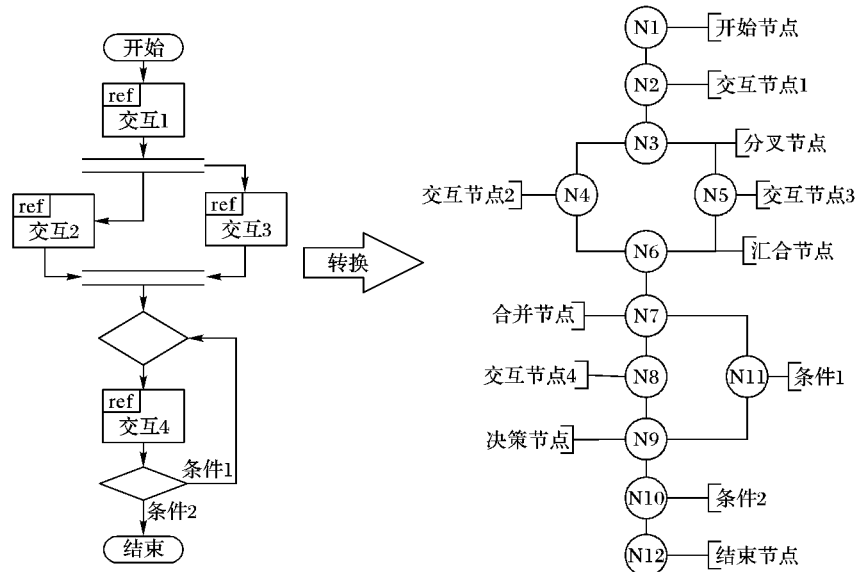


图 2 IOD 活动图结构和 NCFG 的转换

消息时,记为 SDUSF;当为组合框片段时,记为 SDUCF。

本文主要考虑 alt、loop、opt、break 四种类型。图 3~7 展示了顺序图片段到 MSD 之间的转换规则。

UML2.0 顺序图的组合框及其嵌套,使得 UML 顺序

图的结构复杂化。本文规定组合框的嵌套层数不超过 3;loop 的循环次数为 [0,1];alt 根据不同的监护条件执行相应条件区域的消息;opt 当监护条件成立时执行;break 当条件成立时中断退出当前所在组合框。

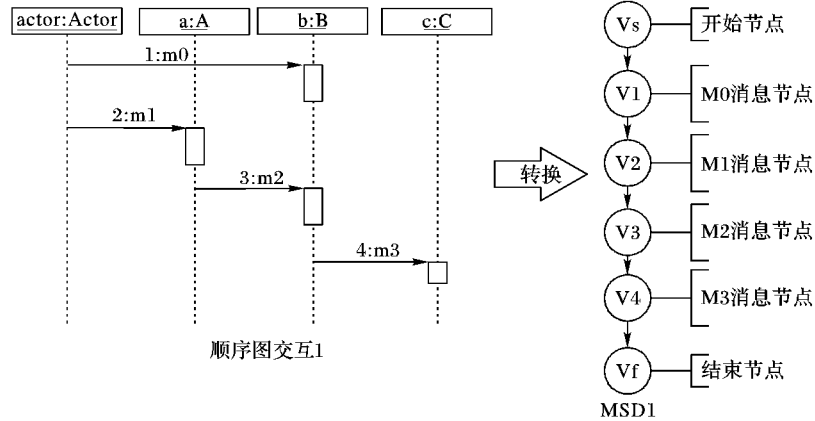


图 3 简单顺序图(SDUSF)

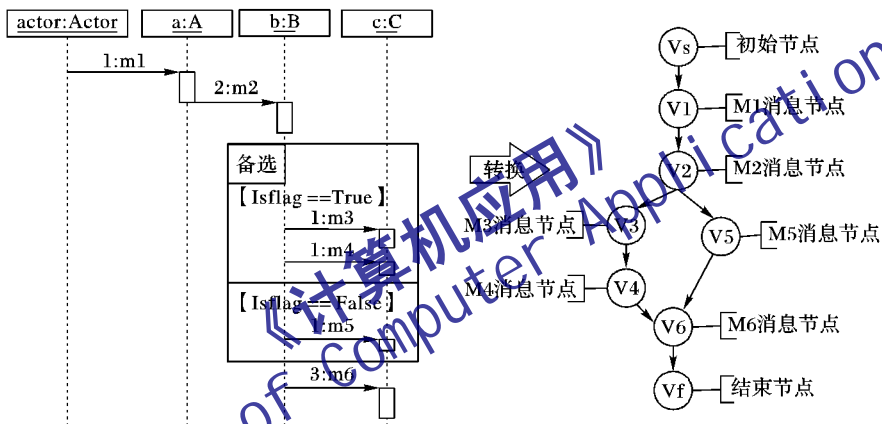


图 4 带有 alt 组合框的顺序图(SDUCF-alt)

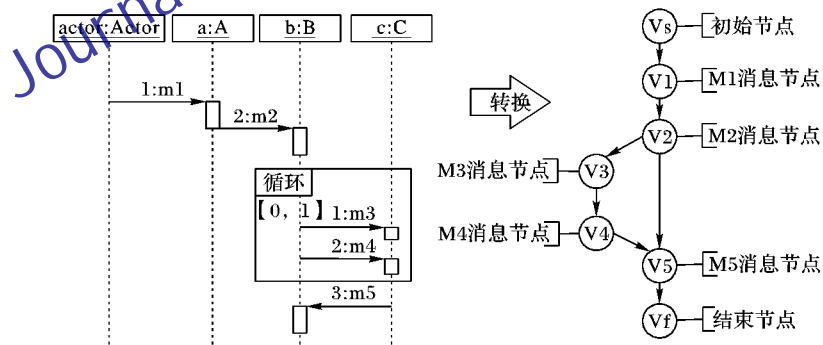


图 5 带有 loop 组合框的顺序图(SDUCF-loop)

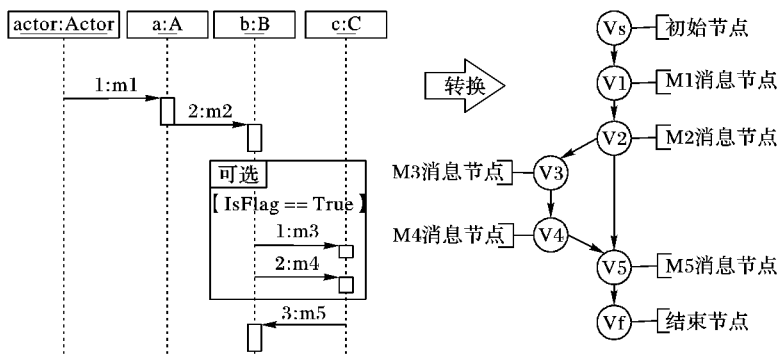


图 6 带有 opt 组合框的顺序图(SDUCF-opt)

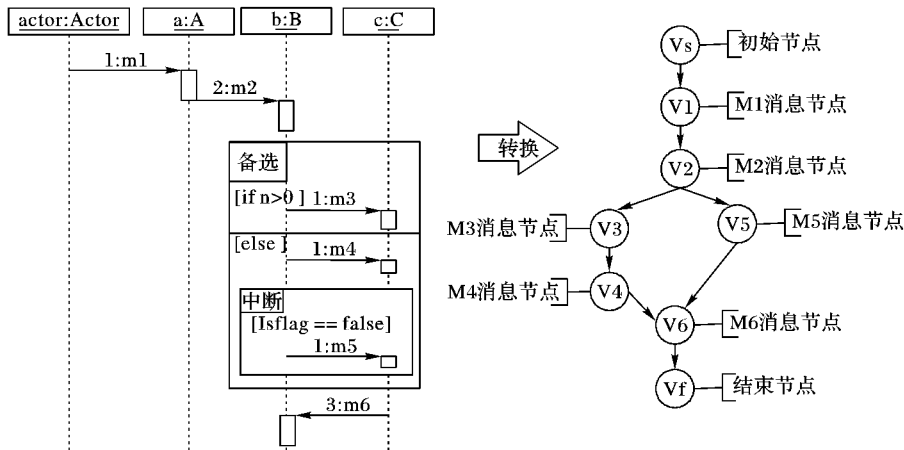


图 7 带有 break 组合框的顺序图(SDUCF-break)

### 3.3 MSD 消息路径提取

消息序列图 MSD 中的消息节点是有序的、不再含有嵌套结构的。通过采用路径覆盖准则获取 MSD 的消息路径 P, 以保证消息路径的高覆盖率和测试的充分性。

根据路径覆盖准则和深度优先算法遍历 MSD, 顺序图片段(图 4~7)获取消息路径如表 1 所示。表 1 中, 不同顺序图中同名的消息节点是不同的, 即 P3 和 P5 是不同的。

表 1 顺序图片段提取消息路径列表

顺序图片段	消息路径 P <sub>i</sub>	路径 P <sub>i</sub> 的具体内容
图 4	V <sub>s</sub> →P1→V <sub>f</sub>	P1 = V1→V2→V3→V4→V6
	V <sub>s</sub> →P2→V <sub>f</sub>	P2 = V1→V2→V5→V6
图 5	V <sub>s</sub> →P3→V <sub>f</sub>	P3 = V1→V2→V3→V4→V5
	V <sub>s</sub> →P4→V <sub>f</sub>	P4 = V1→V2→V5
图 6	V <sub>s</sub> →P5→V <sub>f</sub>	P5 = V1→V2→V3→V4→V5
	V <sub>s</sub> →P6→V <sub>f</sub>	P6 = V1→V2→V5
图 7	V <sub>s</sub> →P7→V <sub>f</sub>	P7 = V1→V2→V3→V6
	V <sub>s</sub> →P8→V <sub>f</sub>	P8 = V1→V2→V4→V6
	V <sub>s</sub> →P9→V <sub>f</sub>	P9 = V1→V2→V4→V5→V6

### 3.4 可测试交互概览图模型构造

可测试交互概览图(Interaction Overview Diagram, IOD)的构造是将 MSD 中提取的消息路径 P 嵌入到 NCFG 对应的交互节点中, 使得构造可测试交互概览图结构相对简单, 同时便于覆盖准则设计。MSD 消息路径 P 是在不同的监护条件下提取的, 它们之间的关系是互斥的, 即每次有且只有一条路径被执行。

将消息路径 P 嵌入到 NCFG 的步骤:

- 1) 深度优先依次遍历 NCFG 的各个节点, 若为交互节点转 2); 否则继续遍历, 直到遍历完整个 NCFG 图。
- 2) MSD 的消息路径 V<sub>s</sub>→P<sub>n</sub>→V<sub>f</sub> 到 V<sub>s</sub>→P<sub>n+k</sub>→V<sub>f</sub> 按照图 8 的表示方法替换 NCFG 中对应的交互节点 N<sub>i</sub>, 执行完毕后转 1)。

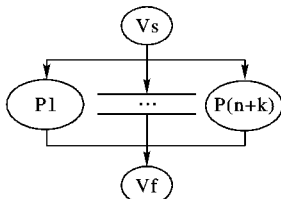


图 8 交互节点的消息路径

按照上述将消息路径嵌入到 NCFG 的算法, 将表 1 中的消息路径分别替换图 2 中 NCFG 中的交互节点 N2、N4、N5、N8, 结果如图 9 所示(V<sub>s</sub> 与 V<sub>f</sub> 之间的路径表示交互节点的消息路径)。

## 4 覆盖准则设计及测试线索生成步骤

### 4.1 覆盖准则设计

可测试的交互概览图的含有分支结构、循环结构、并发结构等复杂结构, 现对这三种主要结构分析并设计覆盖准则。

#### 1) 分支结构。

消息路径的分支和 NCFG 的决策分支构成了 IOD 的分支结构。

覆盖准则 1: 采用路径覆盖准则。每一个分支至少被遍历一次; 分支之间要遵循互斥原则, 尽量使每个分支的覆盖次数保持均衡。

#### 2) 循环结构。

交互概览图的决策与合并导致循环, 主要有 while...do 和 do...while 两种形式。

覆盖准则 2: 采用节点覆盖准则。①对于 while...do 结构, 执行一次循环体即可对所有节点进行覆盖; ②对于 do...while 结构, 需要执行 2 次循环体才可以对所有节点进行覆盖, 当循环体的消息路径有多条互斥分支时, 要求在循环时每次执行循环体所走的路径相同。

#### 3) 并发结构。

IOD 并发结构, 可嵌套 MSD 消息路径, 主要分为不含互斥分支的简单并发结构和含互斥分支的组合并发结构。

覆盖准则 3: ①对于简单并发结构, 为避免组合爆炸和重复测试, 随机选取一条顺序路径进行覆盖。②对于组合并发结构, 在互斥分支中先找出能够并发执行的路径, 再采用两两组合的覆盖测试策略<sup>[14-15]</sup>来进行测试。这种覆盖准则不仅保证了测试的充分性而且有效地避免了组合爆炸。

对于可测试交互概览图中其他复杂结构, 如嵌套消息路径的循环结构和嵌套消息路径并发结构之间的嵌套。处理方案为: ①将被嵌套的结构抽象为一个节点进行处理; ②将被嵌套的结构当作一个子交互概览图处理。

### 4.2 测试线索生成步骤

可测试交互概览图的一条测试线索即是从该图的起始节点到结束节点的一条路径。获取测试线索的步骤如下：

- 1) 将可测试交互概览图进行预处理, 将从并发节点到汇合节点及两个节点间的消息路径看作一个节点。
- 2) 深度优先遍历经过 1) 处理的可测试交互概览图, 根据

相应的覆盖准则处理分支机构和循环结构, 获得初步的测试线索。

- 3) 利用两两组合测试覆盖准则, 构造并发路径, 并随机获取并发路径的一条顺序执行序列; 然后用并发结构的可执行顺序路径替代 2) 中测试线索中的并发节点。

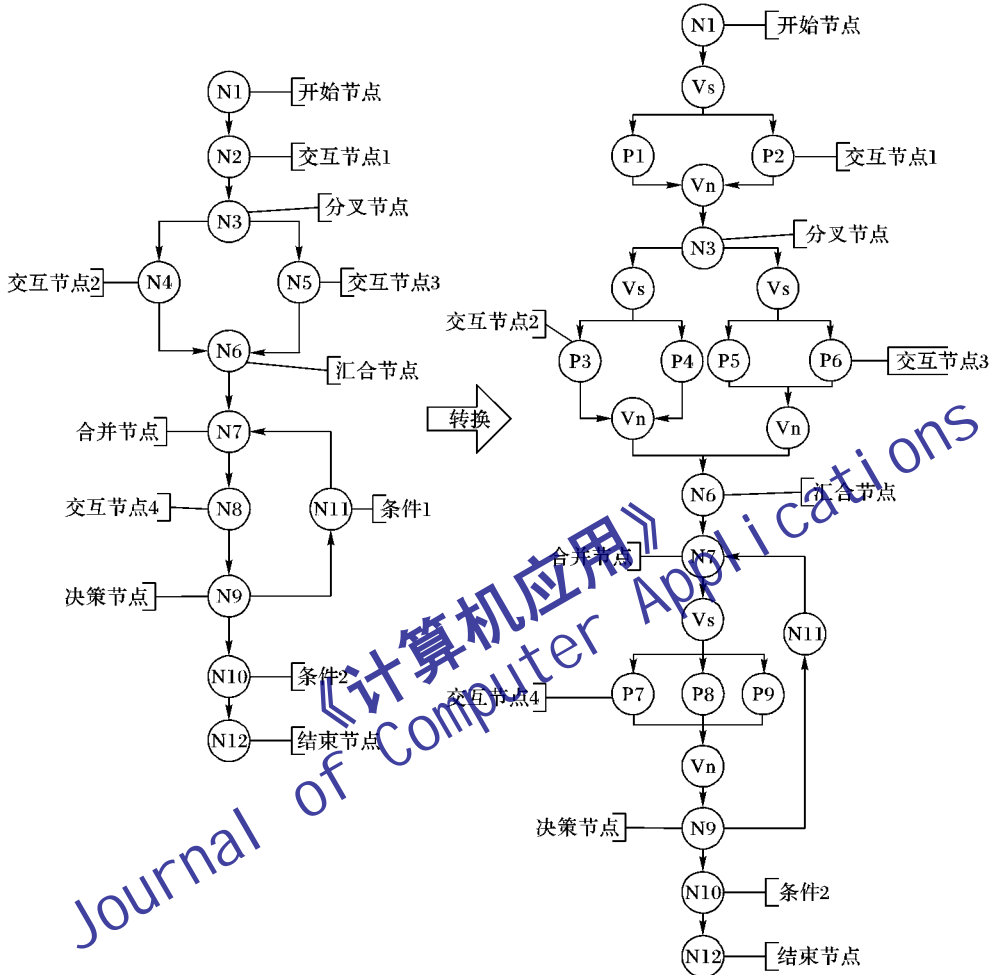


图 9 可测试的交互概览图模型

### 5 实验验证

现以酒店管理系统为例对本文提出的基于 UML 交互概览图测试线索的生成方法进行验证。利用 RSA 进行建模、VS2008 开发环境, .NET 框架, 以 C# 作为开发语言, 来验证所提方法的有效性。

酒店管理系统涉及登录、查看客户请求、处理客户请求、客户入住及退房的管理、房间的检测及清扫管理等功能。

酒店管理系统涉及到的类有 Person、Customer、Employee、Room、customerRecord、customerRequest、loginForm 和 loginControll 八个类, 它们之间的关系如图 10 所示。其中, Customer 和 Employee 继承 Person 类, LoginForm 为边界类, loginControll 为控制类。

酒店管理系统系统的交互概览图中含有一个并发机构和两个循环结构以及 LoginSD、Check Customer Request、Handle Customer Request、Customer Check In、Customer Check Ou、CheckRoom、CleanRoom、BusinessEnd 共 8 个交互节点, 如图 11

所示。

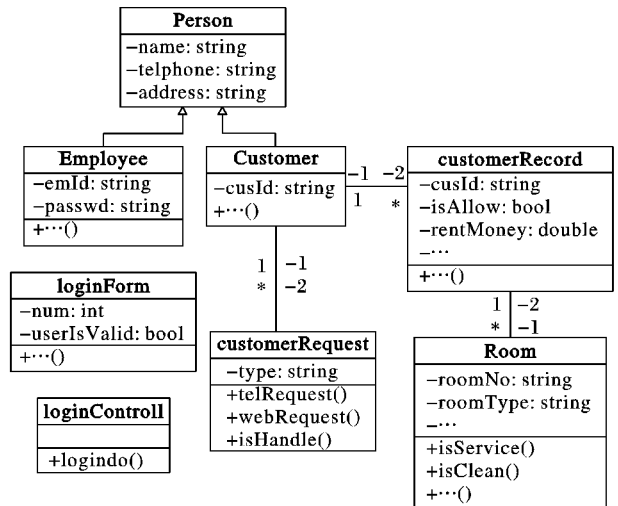


图 10 酒店管理系统类图

该系统的交互节点中涉及到的组合框有 alt、opt、loop、break 4 种结构,现以 LoginSD 工作人员登录顺序图(如图 12 所示)为例展示交互节点的顺序图片段,其中 validate 的作用是用来验证用户名和密码的正确性。

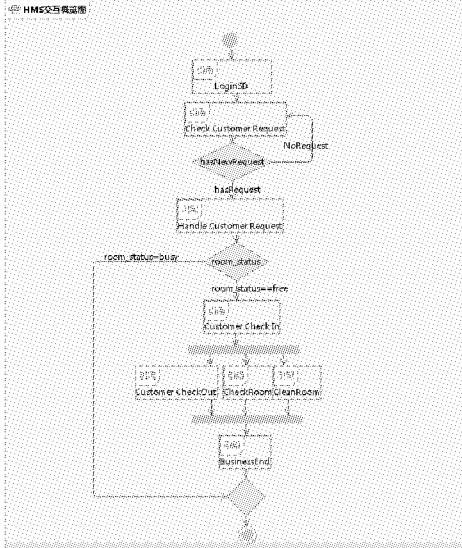


图 11 交互概览图

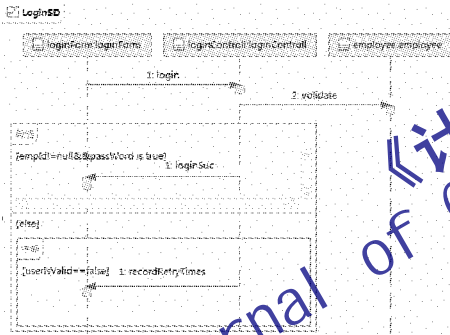


图 12 登录顺序图

对模型文件 XML 进行解析,提取交互概览图的有效信息,根据本文介绍的生成测试线索的流程对其进行分析,最后根据相应的测试覆盖准则提取测试线索。图 13 展示了一条测试线索。

```

符合对象交互信息的可测试交互概览图——两两组合覆盖测试线索 (20 条)
第 1 条: (Initial, InitialNode) -> (null, login() : null, null, loginForm, loginControl) -> (null, validate(empId: String, password: String) : null, null, loginControl, employee) -> (empId: null && password is true) : loginSuc() : null, null, loginForm, loginControl, loginForm) -> (null, checkRequest() : null, null, checkForm, employee) -> ((byCall), reqRequest() : null, null, handleRequest, customerRequest) -> (hasNewRequest, DecisionNode) -> (hasRequest, Condition) -> (null, checkReq() : null, null, checkForm, employee) -> ((byCall), reqRequest() : null, null, handleForm, customerRecord) -> (null, isService, null, null, customerRecord, room) -> (room Status, DecisionNode) -> (free, Condition) -> (null, allow() : null, null, rentForm, customerRecord) -> (null, fillOrder (orderId: String, roomNo: String, rentDate: String, returnDate: String, money: float) : null, null, rentForm, customerRecord) -> (null, updateRoomStatus (status: String) : null, null, customerRecord, room) -> (null, isHandle() : null, null, room, customerRequest) -> (ok, FormNode) -> (null, returnRequest() : null, null, employee, customer) -> (null, getCustomerRecord() : null, null, employee, customerRecord) -> ((else), makeId() : null, null, null, employee, customerRecord) -> (null, checkRoom() : null, null, employee, room) -> (PIS, Construct) -> (Join, JoinNode) -> (PIS, Construct) -> (Merge, MergeNode) -> (Final, ActivityFinalNode)

```

图 13 两两组合覆盖测试线索

根据表 2 给出的结果分析可知,采用两两组合的测试覆

盖准则对可测试交互概览图提取的测试线索能够充分地覆盖整个系统业务,包含模型信息丰富且其总条数比路径覆盖的减少了 16 条(44%),保证测试充分性的前提下避免了组合爆炸的问题。该实验验证了本文算法的可行性与有效性。

表 2 酒店管理系统测试线索生成结果分析

模型	覆盖准则	结果分析
节点控制流程图	路径覆盖	测试线索 2 条,只包含流程信息,测试不完备
消息序列图	路径覆盖	8 个 MSD 测试线索共 13 条,只包含对象交互信息,测试不完备
可测试交互概览图	路径覆盖	测试线索 36 条,测试线索更加全面,包含的模型信息更加丰富
可测试交互概览图	两两组合测试覆盖准则	测试线索 20 条,测试线索数量相对较少,保证了测试的充分性并有效地解决了组合爆炸问题

### 6 结语

本文针对交互概览图的特点,提出了一种结合类图、顺序图、交互概览图自动生成测试线索的方法。该方法考虑了 UML2.0 顺序图中的新特性如组合框 alt、loop、opt、break 等对测试的影响;针对可测试交互概览图的并发、循环、分支结构设计相应的覆盖准则,自动生成的测试线索能够完成对系统的充分测试。通过对酒店管理系统交互概览图的分析,对本文方法进行了验证,生成完备测试线索比路径覆盖产生的测试线索减少了 16 条(44%),有效地避免组合爆炸问题。

今后将进一步研究:1)本文只考虑了同步调用消息以及组合框中的 alt、loop、break 及 opt 对软件测试的影响,后续工作可以对异步消息的调用和其他更多组合框及其嵌套进行处理;2)对模型中的嵌套结构和更加复杂的结构进行研究提出更加优化的覆盖准则,以保证软件测试充分性和提高测试的效率。

### 参考文献:

- [1] JIANG M, WANG T Q, PAN J G. UML state diagram-based test case generation of OO software[J]. Computer Science, 2006, 33(6): 284-290. (江曼, 王天青, 潘金贵. 基于 UML 状态图的面向对象软件测试用例生成[J]. 计算机科学, 2006, 33(6): 284-290.)
- [2] ZHAO X, LIU L L, LUO H P, et al. Method of test scenario generation based on UML dynamic view [J]. Journal of Computer Applications, 2009, 29(5): 1385-1388. (赵欣, 刘琳岚, 罗海平, 等. 一种基于 UML 动态视图的测试场景生成方法[J]. 计算机应用, 2009, 29(5): 1385-1388.)
- [3] SU C C, WANG X J. Research on automatic generating test case method based on UML activity diagram [J]. Computer Technology and Development, 2010, 20(8): 49-55. (苏翠翠, 王晓军. 基于 UML 活动图的测试用例生成方法研究[J]. 计算机技术与发展, 2010, 20(8): 49-55.)

(下转第 291 页)

- Applications, 2012, 32(5): 1470 - 1473. (黄纵, 张晓龙, 边小勇. 基于实时数据流分析的钢铁产品质量监测[J]. 计算机应用, 2012, 32(5): 1470 - 1473.)
- [4] WANG Q, TAO W, DING J, *et al.* Design and realization of real-time data center in power supply enterprise based on PI database [J]. Automation of Electric Power Systems, 2009, 33(6): 99 - 102. (王俏文, 陶文伟, 丁坚勇, 等. 基于 PI 数据库的供电企业实时数据中心的设计与实现[J]. 电力系统自动化, 2009, 33(6): 99 - 102.)
- [5] YANG Y, WANG X, WANG J. Application of real-time database PI in enterprise MES system [J]. Automation & Instrumentation, 2009, 24(12): 38 - 40. (杨源源, 王希宁, 王建华. 实时数据库 PI 在企业 MES 系统中应用[J]. 自动化与仪表, 2009, 24(12): 38 - 40.)
- [6] WANG Z, HUANG T, GE F, *et al.* Application of real-time database PI in wide area measurement system [J]. Electrotechnical Application, 2008, 27(5): 66 - 67. (王正风, 黄太贵, 葛斐, 等. PI 数据库在广域测量系统中的应用[J]. 电气应用, 2008, 27(5): 66 - 67.)
- [7] HWANG Y D. The practices of integrating manufacturing execution systems and six Sigma methodology [J]. The International Journal of Advanced Manufacturing Technology, 2006, 30(7/8): 761 - 768.
- [8] MA L. Six Sigma management [M]. Beijing: China Renmin University Press, 2007: 190 - 206. (马林. 六西格玛管理[M]. 北京: 中国人民大学出版社, 2007: 190 - 206.)
- [9] BAI S, DU F. Parameter optimization and implementation for EWMA control chart based on Markov chain [J]. Manufacturing Automation, 2010, 32(6): 64 - 67. (白双梅, 杜福洲. 基于马尔可夫链的 EWMA 控制图参数优化及其实现[J]. 制造业自动化, 2010, 32(6): 64 - 67.)
- [10] YU L, LIU F. Application of EWMA control chart in stability analysis of MSA [J]. Journal of Systems Engineering, 2008, 23(3): 381 - 384. (俞磊, 刘飞. EWMA 控制图在 MSA 稳定性分析中的应用[J]. 系统工程学报, 2008, 23(3): 381 - 384.)
- [11] TANG S, WANG F. Summary of process capability indices [J]. Chinese Journal of Applied Probability and Statistics, 2004, 20(2): 207 - 208. (汤淑明, 王飞跃. 过程能力指数综述[J]. 应用概率论统计, 2004, 20(2): 207 - 208.)
- [12] CZARSKJ A. Capability process assessment in six Sigma approach [J]. Metallurgy and Engineering, 2007, 33(2): 105 - 112.
- [13] REZAIK, OSTADI B, TAGHIZADEH M R. Applications of process capability and process performance indices [J]. Journal of Applied Sciences, 2006, 6(2): 1186 - 1191.
- [14] CAO Y, ZHU W. Process capability and process performance indices [J]. China Quality, 2005 (1): 77 - 78. (陶泳, 朱文玉. Cpk (过程能力指数) 与 Ppk (过程性能指数) [J]. 中国质量, 2005 (1): 77 - 78.)
- (上接第 275 页)
- [4] LI X Q, CUI J, CAI L Z. Research of software test based on UML statecharts [J]. Computer Engineering, 2011, 37(18): 32 - 34. (李晓庆, 崔健, 蔡立志. 基于 UML 状态图的软件测试研究[J]. 计算机工程, 2011, 37(18): 32 - 34.)
- [5] LI Z Q, SHAO P N, ZHU H. Test cases generation based on UML sequence diagram [J]. Computer Engineering, 2010, 36(22): 58 - 60. (李志强, 邵培南, 朱辉. 基于 UML 顺序图的测试用例生成[J]. 计算机工程, 2010, 36(22): 58 - 60.)
- [6] BOUABANA-TEBIBEL T. Semantics of the interaction overview diagram [C] // IRI 2009: Proceedings of the 2009 IEEE International Conference on Information Reuse & Integration. Piscataway, NJ: IEEE Press, 2009: 278 - 283.
- [7] BOUABANA-TEBIBEL T. UML 2 interaction overview diagram validation [C] // DepCos-RELCOMEX 2009: Proceedings of 2009 the Fourth International Conference on Dependability of Computer Systems. Piscataway, NJ: IEEE Press, 2009: 11 - 16.
- [8] TAN Y J. Elephant — thinking in UML [M]. Beijing: China Water Conservancy and Hydropower Press, 2012. (谭云杰. 大象: Thinking in UML [M]. 北京: 中国水利水电出版社, 2012)
- [9] BLAHA M, RUMBAUGH J. Object-oriented modeling and design with UML [M]. 2nd ed. CHE H, YANG M, translated. Beijing: Posts & Telecommunications Press, 2011. (BLAHA M, RUMBAUGH J. UML 面向对象建模与设计 [M]. 2 版. 车皓阳, 杨眉, 译. 北京: 人民邮电出版社, 2011.)
- [10] MILES R, HAMILTON K. Learning UML 2.0 [M]. WANG Q, translated. Beijing: Tsinghua University Press, 2007. (MILES R, HAMILTON K. UML 2.0 学习指南 [M]. 汪青青, 译. 北京: 清华大学出版社 2007.)
- [11] BARESI L, MORZENTI A, MOTTA A, *et al.* From interaction overview diagrams to temporal logic [M] // Models in Software Engineering. Berlin: Springer-Verlag, 2011: 90 - 104.
- [12] ZENG Y, DING N, ZHOU M. Activity diagram polymorphism test cases generation method with OCL [J]. Computer Engineering, 2012, 38(10): 34 - 37. (曾一, 丁娜, 邹明. 带 OCL 的活动图多态性测试用例生成方法[J]. 计算机工程, 2012, 38(10): 34 - 37.)
- [13] CAO Y, WANG J. UML 2.0 foundation and the modeling tutorial examples of RSA [M]. Beijing: Posts & Telecommunications Press, 2011. (曹衍龙, 王杰. UML 2.0 基础与 RSA 建模实例教程 [M]. 北京: 人民邮电出版社, 2011.)
- [14] FENG Q, WANG L. Research on methods for generating test cases of inter-classes interaction based on UML models and OCL constraints [D]. Zhengzhou: Zhengzhou University, 2012: 38 - 40. (冯秋燕, 王黎明. 基于 UML 模型和 OCL 约束的类间交互测试用例生成方法研究 [D]. 郑州: 郑州大学, 2012: 38 - 40.)
- [15] COHEN D M, DALAL S R, FREDMAN M L, *et al.* The AETC system: an approach to testing based on combinatorial design [J]. IEEE Transactions on Software Engineering, 1997, 23(7): 437 - 444.