构件组装实时系统行为相容性测试用例产生

席 琳[1] 周清雷[1] 李 平[1,2]
(郑州大学信息工程学院 郑州 450001)[1] (信息工程大学电子技术学院 郑州 450004)[2]

摘 要 虽然构件技术在软件开发过程中得到了越来越广泛的应用,但是实时系统是一类设计、实现和验证工作都相当复杂的系统,其构件化远比普通软件复杂,组装仍有许多困难。分析了常见的组装相容性错误,提出了一种实时系统的构件组装行为相容性测试用例产生方法。首先对时间自动机进行扩展,给出了描述实时构件的模型;然后定义了相容性覆盖标准,并把构件行为相容性测试用例生成转化为可被模型检验支持的可达性分析,同时给出了算法;最后用一个实例展示了该方法的具体使用。

关键词 构件,实时系统,行为相容性,测试用例产生,时间自动机
中图法分类号 TP306 **文献标识码** A

Behavioral Compatibility Test Generation of Component-based Real-time System

XI Lin[1] ZHOU Qing-lei[1] LI Ping[1,2]
(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)[1]
(Institute of Electronic Technology, Information Engineering University, Zhengzhou 450004, China)[2]

Abstract Although component-based software development has gained widespread use, composition of the component-based real-time system is still complicated and error-prone. This paper presented an approach for behavioral compatibility test generation of component-based real-time system. Necessary extensions of timed automata (TA) were proposed such that the new model can describe the real-time component. Then, a compatibility coverage criterion was proposed and the behavioral compatibility test generation was converted into reachability analysis of the model, based on which the corresponding algorithm was given. Finally, through an example we demonstrated how our technique works and performs.

Keywords Component, Real-time system, Behavioral compatibility, Test generation, Timed automata

1 引言

基于构件的软件开发方法(CBSD)正在日益广泛地应用于实时领域的系统开发中[1]。实时系统严格限制动作的执行时间,对可靠性有较高的要求,但是,规模大、复杂度高、构件交互行为频繁等这些实时系统的固有特征又常常引起系统开发中构件组装行为不相容等各种错误,从而威胁系统的安全,这个问题引起了许多学者的关注[2-5]。测试是保证软件质量的主要手段,如何精确描述系统并生成有效的测试用例来检验复杂实时构件系统的组装行为,提高系统的可信性,是一个亟待解决的问题。

使用形式化方法生成测试用例的研究一直较为活跃。文献[6]利用模型检验从源代码生成类测试用例。Gargantini和Heitmeyer等提出一种从SCR表示的需求规约生成满足分支覆盖测试标准测试用例的方法[7]。文献[8]介绍了基于UML状态图的测试用例产生方法。文献[9]介绍了可信平台模块满足状态覆盖的测试用例自动生成方法。要寻找被测系统与既定规范不一致的地方,但不可能针对所有的覆盖标准进行测试,因为这样测试成本会变得非常昂贵,并且测试本身也无法在现实环境下实现。另外,实时系统是一类设计、实现和验证工作都相当复杂的系统,其构件化和组装远比普通软件复杂,实时系统需要对组装行为相容性进行特别测试,而以往的测试用例生成方法并未涉及这点。

本文第2节给出对时间自动机[10]的扩展,在此基础上,提出了描述实时构件的模型RCM,并给出了RCM的积的构造方法;第3节分析构件行为相容性问题,给出了基于RCM的构件组装行为相容性测试用例产生方法;第4节给出了应用实例;最后总结全文并指出了下一步研究方向。

2 实时构件的模型

构件是一个具有一定功能的计算或数据逻辑单元。本节在对时间自动机进行扩展的基础上,提出了对构件的交互行为和实时约束特征以及构件体系结构的层次关系进行形式化描述的实时构件模型RCM。

到稿日期:2011-10-27 返修日期:2012-02-17 本文受国家高技术研究发展计划(863)项目(2007AA010408)资助。
席 琳(1981-),女,博士生,主要研究方向为信息安全、形式化方法、软件工程,E-mail:xilinxl@hotmail.com;周清雷(1962-),男,教授,博士生导师,CCF高级会员,主要研究方向为模型检测、安全协议分析、自动机;李 平(1975-),男,博士生,讲师,主要研究方向为信息安全和形式化验证。

为了对构件的交互行为建模，下面引入动作的定义。

定义 1(动作) 动作是构件接收和发出的方法调用事件或者是构件接收和发出通信通道的消息的行为，可分为输入动作、输出动作、内部动作和空动作。空动作表示时间的流逝，而不是方法调用事件或者收发消息的行为。

符号‘!’和‘?’分别表示输出和输入动作。符号‘!’表示方法调用或者向通信通道发送消息，而符号‘?’表示方法被调用或者接收通信通道的消息。

定义 2(实时构件模型 RCM)

实时构件 P 的模型可以表示为一个十一元组：

$\langle ID_P, A_P^I, A_P^O, A_P^H, \Sigma, L, L_0, X, I, E, H\rangle$。其中：

1) ID_P 是构件 P 的标识。

2) A_P^I、A_P^O、A_P^H 是互不相交的 3 个集合，分别表示构件 P 输入动作集合、输出动作集合和内部(隐藏)动作集合；输入动作用来表示被调用的方法或接收通信通道的消息，输出动作用来表示所调用的外界环境方法或其向通信通道发送的消息；内部动作是构件内部的信息交互，对外界不可见。

3) Σ 是一个有穷标记集合，$A_P^I \cup A_P^O \cup A_P^H = \Sigma$。

4) L 是一个有穷位置的集合。

5) $L_0 \subseteq L$ 是一个开始位置集合。

6) X 是一个有穷时钟集合。

7) I 是一个映射，它给 L 中的每个位置 ℓ 指定 $\Phi(X)$ 中的一个时钟约束。

8) $E \subseteq L \times L \times \Sigma \times 2^X \times \Phi(X)$ 是一个转移集合。$\langle \ell, \ell', a!/a?, \lambda, \delta\rangle$ 表示输入动作 $a!/a?$ 时，从位置 ℓ 到位置 ℓ' 的转移。δ 是 x 上的一个时钟约束，它在转移发生时被满足；$\lambda \subseteq X$ 是在该转移发生时复位的时钟集合。

9) H 是该构件的构件组合信息，$H=(P_1, P_2, \cdots, P_n)$ 表示 H 有 n 个元素，构件 P 由构件 P_1、$P_2 \cdots P_n$ 组装而成；特殊情况，$H=(P)$ 表示 H 仅有一个元素。

实时构件模型 A 的语义通过与它相关的转移系统 S_A 的运行来定义。S_A 的一个运行状态是一个二元组 (ℓ, v)，ℓ 是 A 的一个位置，v 是满足不变式 $I(L)$ 的 x 的一个时钟解释。A 的全部运行状态集合为 Q_A，A 的初始状态可表示为 $(\ell_0, 0)$。

对于 S_A 运行中的状态有如下转移类型：

(1)由于时间流逝，无任何动作使状态发生转移。对一个状态 (ℓ, v) 和一个实数的时间增量 $d \geqslant 0$，如果对所有的 $0 \leqslant d' \leqslant d$，$v+d' \in I(\ell)$，则有 $(\ell, v) \xrightarrow{d} (\ell, v+d)$。

(2)由于满足事件要求，有发送动作(或者接收动作)而发生状态转移。对于状态 (ℓ, v) 和转移 $\langle \ell, \ell', a!/a?, \lambda, \delta\rangle$，如果有 $v \in \delta$，那么有 $(\ell, v) \xrightarrow{a!/a?} (\ell', v[\lambda:=0])$。

因此，S_A 是具有标记集 $\Sigma \cup IR^+$ 的转移系统。

实时构件模型中的时钟约束 δ 和复位的时钟集合 $\lambda \subseteq X$ 用于限制实时构件交互行为。

若系统的功能由构件 $P_1, P_2, \cdots, P_n$ 组装实现，则系统模型可以看作对应的实时构件模型的网。(Σ, X) 上实时构件模型的网 $A_1 \parallel \cdots \parallel A_n$ 被定义为 n 个在 (Σ, X) 上的实时构件模型的积。

定义 3(实时构件模型的积)

设 $\langle ID_{P1}, A_{P1}^I, A_{P1}^O, A_{P1}^H, \Sigma_1, L_1, L_{01}, X_1, I_1, E_1, H_1\rangle$ 和 $\langle ID_{P2}, A_{P2}^I, A_{P2}^O, A_{P2}^H, \Sigma_2, L_2, L_{02}, X_2, I_2, E_2, H_2\rangle$ 是实时构件 P_1 和 P_2 的两个模型，假设时钟集合 X_1 和 X_2 不相交，那么构件 P_1 和 P_2 的两个模型的积记为 $A_1 \parallel A_2$，形式如下：$\langle ID_{P1 \parallel P2}, A_{P1}^I \cup A_{P2}^I - \text{common}(P_1 \parallel P_2), A_{P1}^O \cup A_{P2}^O - \text{common}(P_1 \parallel P_2), A_{P1}^H \cup A_{P2}^H \cup \text{common}(P_1 \parallel P_2), \Sigma_1 \cup \Sigma_2, L_1 \times L_2, L_{01} \times L_{02}, X_1 \cup X_2, I, E, (H_1, H_2)\rangle$，其中，$\text{common}(P_1 \parallel P_2) = (A_{P1}^I \cap A_{P2}^O) \cup (A_{P2}^I \cap A_{P1}^O)$，$I(\ell_1, \ell_2) = I(\ell_1) \wedge I(\ell_2)$，积的状态转移集合 E 的构造同 TA 的积的状态转移集合的构造。

3 构件行为相容性测试用例生成

构件行为相容性是指处于同一层次上的若干个组合构件的交互行为没有冲突，没有组合错误。

3.1 常见相容性错误

为了方便查错，首先需要仔细分析构件组合中各种常见的错误，并对这些错误进行归纳分类。

文献[11]根据实时行为协议的特点，定义了 4 种构件相容性错误：停止推进、发散、冗余和无效活动。

停止推进本质上是一种死锁(Deadlock)错误。在基于时间自动机的模型中，可以通过对 $A[]$ not deadlock 的验证表明系统有没有死锁。发散错误本质上是一种活性(Liveness)错误。在基于时间自动机的模型中，系统的活性由转换的时间约束和位置的不变式来保证。冗余在设计阶段对功能模块划分时就可以解决。无效活动指的是构件时间行为协议 P 发出请求(调用)行为，而构件时间行为协议 Q 或没有相应的响应行为，或响应行为事件与请求事件不能满足时效性要求。本文主要对这种行为不相容进行分析。

3.2 不相容的构件行为在构件模型上的形式化表示

当前一个构件中的活动状态的输出动作发生时，另一个构件的活动状态不存在与之对应的输入动作，这两个构件的行为就是不相容的。这种不相容在由构件 P 和 Q 组装的系统模型上表现为：对于行为 a 或者说通道 a，有 $a! \in A_P^O \wedge a? \notin A_Q^I$ or $a! \in A_Q^O \wedge a? \notin A_P^I$。实时系统的行为不相容往往是因为时钟约束的不一致所造成的。

构件模型的网 $A_1 \parallel \cdots \parallel A_n$ 被定义为这 n 个构件模型的并行组装。从语义上来讲，这个网描述了要求所有构件之间延迟变迁同步和离散变迁在互补动作上同步所得的系统。所以，构件行为相容性问题就等价于在由这些构件组装的系统模型上，互补动作是否能在共有通道上同步。

3.3 构件组装行为相容性测试用例生成

好的测试用例可以提高测试的效率，针对构件组装的特点，可以对组装中全部或者某些主要组装行为设计相容性测试用例，以便将来对组装好的系统进行组装相容性的重点测试。本节将模型检验技术用于相容性测试用例生成，并且解释构件相容性测试用例生成问题怎样用有自动工具支持的可达性分析来解决。本文使用工具 UPPAAL[12] 来进行模型检验。

3.3.1 AS-pair

定义 4(AS-pair) 为描述构件行为相容性，假设模型边的数目是可数的，e_i 是边的序号。用 (a, e_i, e_j) 代表行为 a 的 AS-pair，e_i、e_j 分别是动作 $a!$ 和 $a?$ 被定义的边。

如果边 e_i 和 e_j 上定义的互补动作可以同步，那么它们对应的 AS-pair 是有效的。如果某个组装行为对应的所有的

AS-pair 是有效的，那么系统关于这个行为相容。注意，动作 $a!$ 或 $a?$ 可能不止在一条边上定义过，所以要考虑所有的互补动作组合。

AS-pair(a,e_i,e_j)中，a 是行为也是通道，即用于同步的管道，e_i、e_j 分别是动作 $a!$ 和 $a?$ 被定义的边，AS-pair(a,e_i,e_j)是否有效指动作 $a!$ 和 $a?$ 是否能同步，那么，在系统的模型中也就是验证性质 P："有一条路径同时包含 e_i 和 e_j 这两条边"是否成立。在系统的模型中，如果存在一个序列 $s_0\rightarrow s_1\rightarrow\cdots\rightarrow s_n$，使得 s_0 是开始状态，状态之间的转移序列同时包含 e_i 和 e_j 这两条边，则 s_n 就是性质 P 为真的可达状态。这样，AS-pair(a,e_i,e_j)是否有效的问题就转化为可达性验证问题。

3.3.2 相容性测试用例的生成

本节定义相容性覆盖标准并给出基于相容性覆盖标准的相容性测试用例生成方法。

在给出相容性覆盖标准定义之前，需要解决如何用模型的可达性质描述动作相容性问题。上面讨论了 AS-pair(a,e_i,e_j)是否有效的问题可以转化为可达性验证问题；另外，因为某个组装行为对应的所有的 AS-pair 是效的，所以系统关于这个行为相容；因此可以用模型的可达性质描述行为相容性问题。但是，需要先对所有的 AS-pair 进行预处理，方法如下：

(1)为所有的 AS-pair 中不同的边定义一个布尔类型的数组，数组大小为不同边的数目，数组初始值为 false。

(2)对每条边 e_i 赋值 array[i]:=true。

下面给出"行为 A 的 AS-pair 性质"的定义。

定义 5（行为 A 的 AS-pair 性质） 所有的 AS-pair 进行预处理后，如果 A 对应的 AS-pair 是(a,e_i,e_j)，那么行为 A 的 AS-pair 性质即为 $E\langle\rangle$ (array[i]= =true and array[j]==true)。

在以上定义的基础上，给出相容性覆盖标准的定义。

定义 6（行为 A 的相容性覆盖） 一组测试用例满足针对组装行为 A 的相容性覆盖，当且仅当本组测试用例在模型上执行时有可达状态满足行为 A 的 AS-pair 性质 P。可以看出，测试用例满足行为 A 的相容性覆盖，当且仅当它是 $E\langle\rangle P$ 的证据路径集。

每个组装行为可能不止一个 AS-pair，因此，需要对每个行为的所有 AS-pair 产生测试用例。

满足关于行为 A 的相容性覆盖的测试用例产生方法如下：

1)确定待测行为 A 对应的 AS-pair；

2)定义待测行为 A 对应的 AS-pair 的"陷阱"性质，即 E〈〉(array[i]= =true and array[j]== true)的非；

3)利用模型检验工具对给定性质产生诊断路径的功能，在模型检验中捕获相应的反例，而这些反例就是满足相应组装行为相容性的测试用例。

对组装中的全部行为也可以产生测试用例，它是关于行为 A 的相容性覆盖测试用例产生方法的扩展，下面给出产生算法，输入是各构件模型根据定义 3 得到的组装的实时构件模型，输出是相容性验证结果和测试用例。

输入：组装的实时构件模型$\langle ID_P,A_P^I,A_P^O,A_P^H,\Sigma,L,L_0,X,I,E,H\rangle$

输出：相容性验证结果和测试用例

(1) for (E 中的每个迁移：$\langle\ell,\ell',a!/a?,\lambda,\delta\rangle$)

(2) 标记边序号 $e_i\in$ Edge

(3) for (每个 $a\in A_P^H$)

(4) for (每个 a! 被定义的边 e_i)

(5) for (每个 a? 被定义的边 e_j)

(6) 生成 AS-pair(a,e_i,e_j)，并加入集合 AP

(7) for (每个 $e_i\in$ Edge)

(8) array[i]:=true

(9) for (每个 AS-pair$\in$AP)

(10) 验证！E〈〉(array[i]= =true and array[j]==true)

(11) if 返回 false 输出诊断路径

(12) else 输出 H 中构件组装行为不相容信息

4 实例分析

本节通过组装一个简单的单机单衡式称重管理系统来演示以上的方法。

系统满足的规则如下：

• 磅房每次只允许一个货车上磅，上磅称重的标准有两种。

• 货车进入等候区时，货车配备的带有智能 IC 卡的终端 HHT 发出到达等候区的信号 approach。控制器收到信号后判断电子衡器是否空闲，如果已经被占用，立即向车载终端 HHT 发送停车等候消息 stop，货车在等候区等待；否则，控制器发出过磅信息给电子衡器，并在不超过 10s 的时间内给车载终端 HHT 发出货车上磅信息 go。

• 电子衡器接收到从控制器发送的相应过磅信息 weight 后在 10s 的时间内从数据库中提取对应该卡号的信息，将信息填写到称重界面的相应编辑框并进入相应标准的产品称重模式，发出就绪信号 ready。

• 货车通过电子衡器的时间不超过 5s，当它下磅后，车载终端 HHT 向控制器发出离开信号 exit。控制器收到信号 exit 后，如果接下来还有货车等待过磅，便发出相应的过磅信息，在不超过 10s 的时间内发出 go 信息给等待的货车；如果没有货车等待过磅，控制器就回到空闲状态。

• 对多个货车的过磅请求，遵循先来先服务原则进行调度。

从构件库中选择实现控制器、车载终端、电子衡和队列功能的构件。对它们建模，得到系统的成员构件模型分别为：HHT(车载终端)、Scale(电子衡)、Controller(控制器)和 Queue(队列)，表示如下：

$C1=$(HHT,{stop,go},{appr,exit},{∅},{stop? ,go? ,appr! ,exit! },{Start,Appr,Wait,Cross,Exit},{Start},{x},$I,E,(C1)$)，其中 I,E 等的信息如图 1 所示。

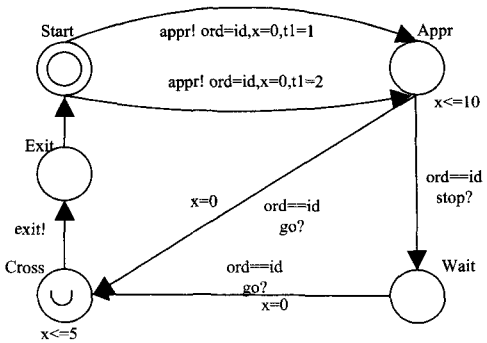

图 1 HHT 的时间自动机模型

$C2=$(Scale,{weigh1,weigh2},{ready},{Ø},{weigh1?,weigh2?,ready!},{Rule1,Rule2,Turn1,Turn2},{Rule1},{z},$I,E,$($C2$)),其中 I,E 等的信息如图 2 所示。

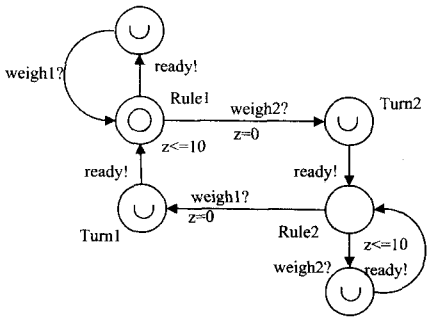

图 2　Scale 的时间自动机模型

$C3=$(Controller,{empty,notempty,appr,ready,exit},{add,weigh1,weigh2,gt,stop,gid,go,rem},{Ø},{empty?,notempty?,appr?,ready?,exit?,add!,weigh1!,weigh2!,gt!,stop!,gid!,go!,rem!},{idel,notem,turn,cross,idle},{idel},{t},$I,E,$($C3$)),其中 I,E 等的信息如图 3 所示。

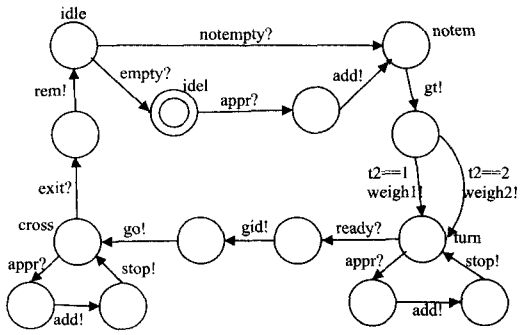

图 3　Controller 的时间自动机模型

$C4=$(Queue,{add,gt,gid,rem},{ empty,notempty },{Ø},{add?,gt?,gid?,rem?,empty!,notempty!},{idel,shift},{idel},{Ø},$I,E,$($C4$)),其中 I,E 等的信息如图 4 所示。

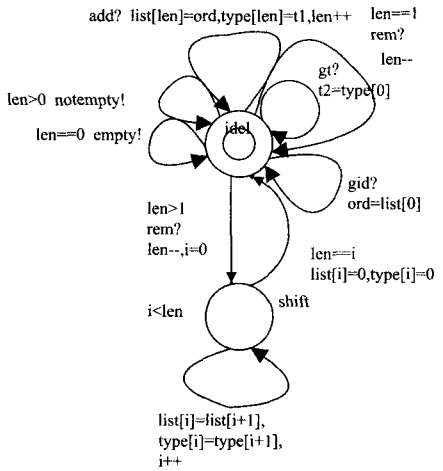

图 4　Queue 的时间自动机模型

用 UPPAAL 可以自动构造以上构件模型转换系统的积:HHT ‖ Scale ‖ Controller ‖ Queue,即 $H=(C1,C2,C3,C4)$。系统的内部动作 $A^{I}_{HHT\,|\,Scale\,|\,Controller\,|\,Queue}$ 即为 13 个通道:appr,go,exit,stop,ready,weigh1,weigh2,add,rem,empty,notempty,gt,gid。

下面针对构件 HHT 和构件 Controller 关于动作 stop 的行为相容性设计测试用例,以便在将来的测试阶段进行重点测试。

采用 3.3 节定义的方法,首先,确定待测动作 stop 对应的 AS-pair:(stop,e_{23},e_3)和(stop,e_{29},e_3),其中,e_{23} 和 e_{29} 代表 stop! 被定义的边,e_3 是 stop? 被定义的边,stop! $\in A^{O}_{Controller}$,stop? $\in A^{I}_{HHT}$。然后,分别将 E〈〉(array[23]= =true and array[3]== true)的非和 E〈〉(array[29]= =true and array[3]== true)的非作为"陷阱"性质,利用模型检验工具得到满足相应组装行为相容性的测试用例。

图 5 是在 UPPAAL 的模拟器中随机得到的证据路径,也就是测试用例,它是一个各实体之间通过管道相互通信、控制的消息序列(MSC)。

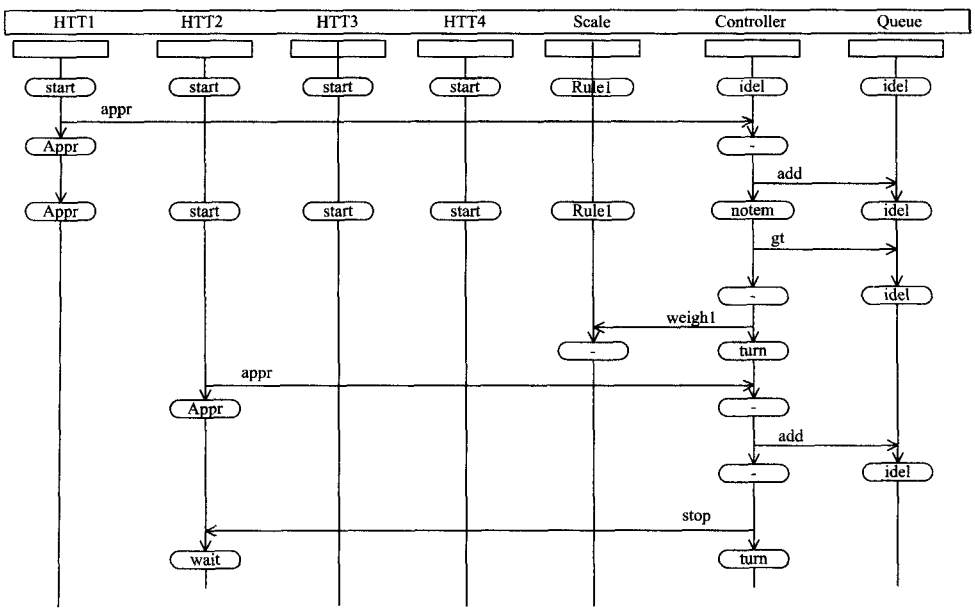

图 5　测试用例 1 的 MSC 图

图 6 是在 UPPAAL 的模拟器中得到的另一个测试用例　　的 MSC 图。

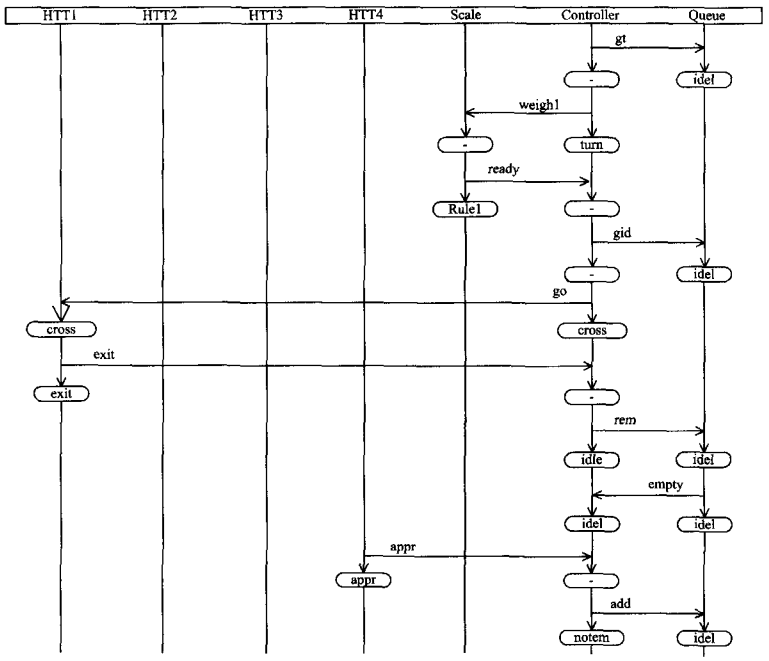

图 6　测试用例 2 的 MSC 图

这些测试用例可以对容易出错的组装行为重点测试，大大提高了测试工作的有效性。

结束语　本文在对时间自动机进行扩展的基础上，提出了描述实时构件的模型 RCM，并基于此给出了一种实时构件行为相容性测试用例产生方法。构件模型 RCM 可以在构件系统的分析设计中提供构件行为实时属性和层次特性的形式化描述，以减少开发人员对系统行为理解上的歧义。行为相容性测试用例产生方法定义覆盖标准，并利用模型检测方法对组装动作设计相容性测试用例。由于模型检测工具的支持，因此本文方法可以实现自动化。

下一步将开展构件实时行为建模、验证、测试等相关技术和实用工具的开发实现研究。

参 考 文 献

[1] Alagar V, Mohammad M. A component model for trustworthy real-time reactive systems development[C]// Formal Aspects of Component Software (FACS'07). Sophia-Antipolis, France: ENTCS, Elsevier, Sep. 2007: 1-15

[2] Kofron J. Checking Software Component Behavior Using Behavior Protocls and Spin[C]// Proceedings of the 2007 ACM Symposium on Applied Computing. New York: ACM Press, 2007: 1513-1517

[3] 贾仰理，张振领，李舟军. 基于自动机的构件实时交互行为的形式化模型[J]. 计算机科学，2010，37(9)：151-156

[4] 毛斐巧，齐德昱，林伟伟. 一种解决构件连接死锁问题的方法[J]. 软件学报，2008，19(10)：2527-2538

[5] 金仙力. 实时服务构件的语义特征和行为组装形式化技术研究[D]. 北京：北京邮电大学，2007

[6] 梁陈良，聂长海，徐宝文，等. 一种基于模型检验的类测试用例生成方法[J]. 东南大学学报，2007，37(5)：776-781

[7] Gargantini A, Heitmeyer C. Using model checking to generate tests from requirements specifications [C]// Proceedings of the 7th European Engineering Conference Held Jointly with the 7th ACM'S- IGSOFT International Symposium on Foundations of Software Engineering. Toulouse, FR, 1999: 146-162

[8] 章涛，顾庆，陈道蓄. 基于 UML 状态图的测试技术研究[J]. 计算机科学，2007，34(10)：264-280

[9] 陈小峰. 可信平台模块的形式化分析和测试[J]. 计算机学报，2009，32(4)：646-653

[10] Alur A W, Dill D L. A theory of timed automata [J]. Theoretical Computer Science(S0304-3975), 1994, 126(2): 183-235

[11] 贾仰理，张振领，李舟军. 构件行为协议实时性扩展及相容性验证 [J]. 计算机科学，2010，37(10)：143-147

[12] Bengtsson J, Larsson F, Larson F, et al. UPPAAL-a Tool for Automatic Verification of Real-time Systems[C]// Proceedings of the DIMACS/SYCON workshop on Verification and control of Hybrid systems Ⅲ. Springer-Verlag, New York, October 1995: 232-243

(上接第 104 页)

[2] 王健，王慧强，赵国生. 基于不确定型 AHP 的网络生存能力模糊综合评估[J]. 计算机科学，2006，33(6)：73-75

[3] 刘啸林，王能. 通信网络抗毁性量度研究[J]. 上海师范大学学报：自然科学版，2006，35(5)：38-41

[4] 明亮，王东霞，张鲁峰，等. 网络抗毁性测度研究[J]. 计算机应用研究，2010，27(5)：1850-1852

[5] 王志刚，杨世松. 基于模糊综合评价的信息网络系统可生存性评估[J]. 网络安全技术与应用，2010，10：58-60

[6] 李黎，管晓宏，赵千川，等. 网络生存适应性的多目标评估[J]. 西安交通大学学报，2010，44(10)：1-7

[7] 纪震，廖惠连，吴青华. 粒子群算法及应用[M]. 北京：科学出版社，2009

[8] 郭虹，兰巨龙，刘洛琨. 考虑节点重要度的 Ad Hoc 网络抗毁性测度研究[J]. 小型微型计算机，2010，31(6)：1063-1066