TCBV:一种构件时序行为建模与相容性验证工具

张振领[1] 贾仰理[1] 周恩光[2] 李舟军[2]
(聊城大学计算机学院 聊城 252059)[1] (北京航空航天大学计算机学院 北京 100191)[2]

摘 要 利用形式化方法对复杂实时构件系统的时序行为进行建模与验证对于提高安全攸关实时构件系统的正确性、可靠性与安全性具有重要意义。介绍了基于时间行为协议的构件时序行为的形式化建模和相容性验证方法,给出了时间行为协议建模与相容性验证工具TCBV的系统架构与功能模块。TCBV应用方便,能够实现实时构件时序行为模型的图形化表示,并可对复杂交互行为的相容性进行自动验证。结合应用实例,介绍了如何利用TCBV对复杂实时构件系统的时序行为进行建模和验证。最后,将TCBV与其它相关工具进行了比较。

关键词 实时构件系统,时序行为,形式化方法,建模,相容性验证工具
中图法分类号 TP311 **文献标识码** A

Specification and Compatibility Verification of Timing Behavior in Components

ZHANG Zhen-ling[1] JIA Yang-li[1] ZHOU En-guang[2] LI Zhou-jun[2]
(School of Computer, Liaocheng University, Liaocheng 252059, China)[1]
(School of Computer Science & Engineering, Beihang University, Beijing 100191, China)[2]

Abstract It can efficiently improve the system's correctness and reliability for specification and verifying of timing behavior in complex real-time components. This paper presented the timed behavior protocol based formal modeling methods of timing behavior and the verification method for component-based systems. The architecture of the specification and verification tool named TCBV was given. An application example was introduced and the experimental results show that the timed behavior protocol based specification and verification methods can accurately model and conveniently verify errors of timing behavior in complex real-time component systems. Finally, TCBV and other related tools were compared and the differences between them were analyzed.

Keywords Real-time component system, Timing behavior, Formal methods, Specification, Compatibility verification tool

1 引言

随着计算技术、通信技术和自动控制技术的飞速发展和成熟,信息物理融合系统(Cyber-Physical System,CPS)开始在世界范围内出现。很多CPS系统是一种复杂的实时构件系统,在航空航天、医疗、交通等领域的应用日趋广泛。这类CPS构件系统共同的特点是对可靠性、安全性等要求特别高,系统失效可能会造成极大的生命、经济损失。因此,系统必须满足可靠性和安全性,其行为必须具备可预测性和自适应性,其操作必须具有可靠性和可验证性[1]。

基于形式化描述和验证方法对复杂CPS构件系统的交互行为进行建模与验证可以有效提高系统的正确性、可靠性和安全性。通过形式化描述和验证等方法,可以确定复杂CPS系统的边界,更精确地刻画CPS构件系统的时序行为和各种非功能性需求,避免不同阶段开发人员对系统的理解歧义,并为系统的正确性、可靠性和安全性等各种可信性质的验证提供自动化验证手段,极大地减少系统设计开发早期阶段的错误,并避免后期错误修改所带来的高额成本开销。因此,在实际应用中,对这类复杂构件系统的交互行为进行建模与验证具有重要的意义。

在文献[2,3]中,分别对基于模型检验的构件验证技术、基于时间行为协议的实时构件行为相容性错误检测方法进行了分析探讨。本文在此基础上,针对CPS实时构件系统的快速发展和建模与验证的需求,对基于时间行为协议的构件时序行为建模与验证理论进行了分析和研究,介绍了开发的构件时序行为建模与验证工具,并结合实际给出了应用示例。

本文第2节介绍了基于时间行为协议的时序行为建模与组合相容性验证方法;第3节介绍了时间行为协议的建模与验证工具TCBV的框架与功能模块,结合验证工具给出了应用实例;第4节将TCBV与其它相关工具进行了比较;最后总结全文并指出下一步的工作。

到稿日期:2012-05-08 返修日期:2012-08-10 本文受国家自然科学基金项目(90718017),山东省自然科学基金项目(ZR2011FL023),山东省软科学项目(2010RKE16007),山东省高校智能信息处理与网络安全重点实验室项目资助。
张振领(1977—),女,硕士,讲师,主要研究方向为形式化方法、智能信息处理,E-mail:zhangzhenling@lcu.edu.cn;**贾仰理**(1976—),男,博士,主要研究方向为形式化方法等;**周恩光**(1984—),男,博士生,主要研究方向为信息安全等;**李舟军**(1963—),男,博士生导师,主要研究方向为形式化方法、信息安全、智能信息处理等。

2 基于时间行为协议的建模及相容性验证方法

常见的形式化描述方法包括基于进程代数的方法、基于自动机的方法、基于 Petri 网的方法和基于逻辑的方法等。时间行为协议是基于进程代数的方法的一种，它把并发执行的进程的行为看成是各个单一进程的行为的所有可能的交错合成，可以给出并发过程的显式表示。其对系统(子系统、构件)行为的刻画较为直观和细致，同时又可以对系统的状态及其演变进行表达和推理，并且易于融合到系统接口、模块等的形式化规约。

2.1 时间行为协议简介

行为协议[4]由事件标记和操作符组成：事件标记用于表示构件中方法的调用，操作符用于构造复杂的行为协议。事件的表示由接口名、方法名、事件类型组成。事件类型由符号“!”、“?”、“τ”、“↑”、“↓”表示，其中事件前缀“!”表示发出(emit)事件，“?”表示接收(accept)事件，“τ”表示内部事件；事件后缀“↑”和“↓”分别表示“请求(request)”和“响应(response)”。可以利用行为协议来描述复杂构件各个层次上的行为，给出所描述构件提供(provide)或请求(require)的原子事件(方法调用)的所有可能序列。例如，数据库构件接收调用，依次启动日志和事务处理行为可表示为：? db. start↑{! logger. start↑;! tm. init↑}。行为协议定义简单，易于阅读和书写，并且支持行为描述的逐步求精，因此一经提出立刻得到了广泛关注。由于行为协议定义时未考虑时间因素，因此，行为协议对时间因素的描述能力先天不足，难以满足对实时系统、嵌入式和分布式系统的描述和验证要求。

文献[3]在行为协议语法的基础上提出了时间行为协议 TBP(Timed Behavior Protocol)。TBP 引入时间概念，对一些事件绑定时间限制属性，并引入与时间相关的操作符。在该模型下，事件的时间是单调递增的无界实数，通过设置时钟变量，可分析与比较不同事件发生的时机，并可对时钟变量进行赋值、复位等操作。

时间行为协议引入了时间事件的概念来描述实时构件发送、接收、请求、响应情况的时间属性，其具体表示形式和对应的含义如下：

! interface. method[t1]↑表示调用方构件在时刻 t1 内发出接口 interface 上的 method 方法的调用请求；

? interface. method[t2]↑表示接收方构件在时刻 t2 内接收到接口 interface 上的 method 方法的调用请求；

! interface. method[t3]↓表示接收方构件在时刻 t3 内发出接口 interface 上的 method 方法的调用响应。

? interface. method[t4]↓表示在调用方构件在时刻 t4 内接收到接口 interface 上的 method 方法的调用响应；

简单的时间行为协议可以仅包含一个事件，为了表示构件复杂的行为，文献[2]定义了组合操作符来构造更复杂的时间行为协议。设 A,B 为时间行为协议，t 为时间，TBP 语法子集的巴克斯范式定义如下：

$$P::=Reset(t)\mid Idle(t)\mid \mathrm{Stop}\mid \mathrm{Skip}\mid \mathrm{Error}\mid P;Q\mid P+Q$$
$$\mid P^{*}\mid P\triangleright_{t}Q\mid P\mid Q\mid P\parallel Q\mid P/G\mid P\bigcap\nolimits_{x}Q\mid$$

式中，*Reset*、*Idle* 等是特殊类型的事件，$Reset(t)$ 表示时钟变量 t 的复位事件；$Idle(t)$ 表示一段时间的流逝；Stop 表示终止事件；Skip 表示跳过事件；Error 表示错误事件。更多时间行为协议的知识请参考文献[3]。

关于时间系统的建模语言还包括 Hybrid CSP[5]、XYZ/E[6]、混成 I/O 自动机[7]以及时间 Petri 网、基于 UML 的时间统一建模语言 TUML 等。与时间行为协议相比，这些方法存在理论复杂、一般程序设计人员不易掌握、建模独立于构件模型等缺点。时间行为协议能够无缝融合到构件模型，并且具有语法和语义简单、方便软件开发人员掌握的特点。

2.2 时间行为协议的组合行为的建模

设两个时间行为协议 P,Q，它们组合形成的时间事件序列为 P、Q 事件标识的交叠序列，要求在给定事件集合 X 上同步。在组合操作下，需要考虑 P 与 Q 之间的交互。假设 A 和 B 分别为时间行为协议 P 与 Q 的事件集合。在交互情况下，时间行为协议 P 或 Q 执行 send a 事件，时间行为协议 Q 或 P 则执行 receive a 事件，时间行为协议 P 与 Q 对事件进行交互，在组合操作下，则生成内部事件。即在满足时序要求的情况下，任意子序列! interface. method[t],? interface. method[t]或? interface. method[t],! interface. method[t](? interface. method[t]$\in X$,! interface. method[t]$\in X$)组合为 τinterface. method[t]。

即时间行为协议的组合序列为两协议形成事件序列的交叠序列。对指定事件集合内的事件，该交叠序列需要完成发出事件(!)和接收事件(?)的匹配，生成内部(τ)事件，如果部分事件 $x\in X$ 没有完成匹配，则从组合后的路径中删除。

在理想情况下，事件 $x\in X$ 完全匹配，即时间行为协议 P 或 Q 行为序列执行 send a 事件，时间行为协议 Q 或 P 相对应的行为序列正好执行 receive a 事件，则两时间行为协议 P 与 Q 对事件进行交互，生成内部事件。除特殊情况外，可以不考虑内部事件的时间要求。

例如：

P=! tm. begin[1];(! tm. commit[1]+! tm. rollback[1])

Q=? tm. begin[1];? tm. rollback[1]

则 P 和 Q 在集合 X={begin,commit,rollback}上的组合为：

$P\bigcap_{x}Q=\tau$tm. begin;τtm. rollback

组合后的时间行为协议不包含事件 tm. commit，这是因为组合操作符右边要求必须出现 rollback 事件，而 rollback 和 commit 事件只能选择其一执行。

2.3 相容性验证

在时序行为形式化描述的基础上，可以对构件之间、构件和外部环境(体系结构)之间的组合行为进行建模。在模型基础上可以进行行为相容性分析和验证研究。一个复杂的实时构件系统需要由多个子系统组合而成，我们需要分析参与组合的各个子系统(子构件)之间的行为是否相容。

例如，常见的一种相容性错误是：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。文献[3]给出了常见的组合相容性错误，本节在此基础上提出相

容性检测方法，这是实现相关验证工具的基础。

我们定义操作符$\nabla_t\{X\}$来对时序行为的相容性进行检测。

设X为方法集合，对于时间行为协议P和Q，$L(P\ \nabla_t\{X\}Q)$为动作集合X上P和Q的所有可能的同步交叠路径(interleaved trace)。同步是指对于每一个$m\in X$，任意形式为$?m[t]\uparrow$或$?m[t]\downarrow$的事件都会等待相应的调用事件$!m[t]\uparrow$或$!m[t]\downarrow$，并组合为$\tau m\uparrow$或$\tau m\downarrow$的形式。$L(P\ \nabla_t\{X\}Q)$形成的路径可能包含文献[3]中介绍的几类常见错误。用$\mathrm{ERROR}m\uparrow$或$\mathrm{ERROR}m\downarrow$来表示无效行为$m$，用STOP和$\varepsilon\infty$分别表示停止推进和发散动作。

例如，

$P=!a[1]\{?m[1]\}+!b[1]\{?n[1]\}$

$Q=?a[1]\{!m[1]+!n[1]\}+?b[1]\{!n[1]\}$

给出它们执行∇_t操作的结果：

$L(P\ \nabla_t\{X\}Q)=\{\langle\tau a\uparrow,\tau m\uparrow,\tau m\downarrow,\tau a\downarrow\rangle$

$\langle\tau a\uparrow,\mathrm{ERROR}n\uparrow,\tau a\downarrow\rangle,\langle\tau b\uparrow,\tau n\uparrow,\tau n\downarrow,\tau b\downarrow\rangle\}$

式中，集合$X=\{a\uparrow,a\downarrow,b\uparrow,b\downarrow,m\uparrow,m\downarrow,n\uparrow,n\downarrow\}$。

时间行为协议P、Q分别定义如下：

$P=!a[2]\{?m[1]+!b[1]\}$

$Q=?a[1]\{!m[1]+?b[1]\}$

在集合$X=\{a\uparrow,a\downarrow,b\uparrow,b\downarrow,m\uparrow,m\downarrow,n\uparrow,n\downarrow\}$上组合时，结果为：

$$L(P\ \nabla_t\{X\}Q)=\mathrm{ERROR}a\uparrow\{\tau m\uparrow,\tau m\downarrow\}+\mathrm{ERROR}a\uparrow\{\tau b\uparrow,\tau b\downarrow\}$$

而P、Q分别如下定义时：

$P=?m[1];?n[1]$

$Q=!m[1];?a[1]$

当它们在集合$\{a\uparrow,a\downarrow,m\uparrow,m\downarrow,n\uparrow,n\downarrow\}$上组合时，结果为：

$L(P\ \nabla_t\{X\}Q)=\tau m[1]\uparrow,m[1]\downarrow,\mathrm{STOP}$

因此，为了更好地表示构件组合错误，在构件组合操作$\cap_x$的基础上定义了操作符∇_t，显然：

$L(P\ \nabla_t\{X\}Q)=L(P\cap_x Q)\cup\mathrm{Errortrace}L(P,Q)$

式中，$\mathrm{Errortrace}L(P,Q)$表示包含无效行为、停止推进或发散等错误的行为迹。

对于同一层次上的两个组合构件的时间行为协议，可以利用$\nabla\ t\{X\}$进行相容性检测。如果时间行为协议组合后产生的路径包含错误事件，则可以判断两个构件行为存在相容性错误，不能组合。

3 TCBV——基于时间行为协议的建模与验证工具

为更好地支持复杂实时构件系统的开发，在相容性检测理论基础上，我们开发了支持时间行为协议建模和自动化验证的工具。

3.1 TCBV基本框架与功能模块

我们基于模型检验理论开发了时间行为协议组合验证工具TCBV(Timed Component Behavior Verifier)，该工具实现了构件时间行为协议的输入、编辑、图形显示、行为组合、精化功能，并支持对行为协议进行相容性分析验证，把基于构件的可信组合的形式化描述方法和构件组合的可信性质验证相结合，从理论和实践角度验证了构件组合的可信性。

系统的基本框架如图1所示。

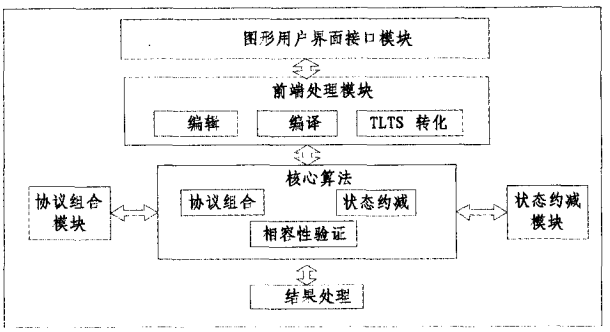

图1 TCBV系统框架图示

工具系统按照功能实现可以划分为前端处理模块和核心算法模块，其中前端处理模块实现时间行为协议的输入、编辑、TLTS转化和图形显示，实现核心算法部分为构件时序行为协议的组合、约简及验证提供支持。

前端处理模块：支持时间行为协议的输入、保存、导入、导出、编辑操作；支持时间行为协议的编译，能给出各种错误提示；支持将时间行为协议用标记迁移系统给出图形化显示，方便观察和查错。

协议组合模块：支持时间行为协议的组合操作，对两个或多个时间行为协议能够给出其组合后的结果。本模块是基于时间行为协议的组合序列理论实现的。

协议约简模块：支持时间行为协议的精简(最小化)，将一些无用的状态去除。约简协议一般仅能够产生被约简协议在特定应用中的部分行为迹，这些迹往往都是由外围环境触发，约简协议可以代替被约简协议来响应这些触发事件。在本模块中，为了验证组合中的两个时间行为协议是否可以约简，首先把组合中用到的事件标识出，标识的事件肯定不能削减掉，而其它的则可以考虑是否可以约简。分别考察这些协议片断是否可产生空迹或能否直接发出调用事件，然后根据情况判定是否可以将这些协议片断约简。

验证模块：支持对组合的时间行为协议验证是否包含“bad activity”等组合错误；支持对时间行为协议进行死锁、安全性、活性等的验证。本模块的相容性验证算法参见文献[3]。

可以将TCBV应用于系统开发的设计阶段，对设计者通常所关心的一些构件相容性等有关性质进行有效的分析和验证。

3.2 应用示例

为表示和图示的方便，下面以文献[3]中的一个简化的锅炉压力控制构件系统的验证为例，说明如何利用开发的工具TCBV进行基于时间行为协议的时序行为建模与验证。

考虑一个简化的锅炉压力控制构件系统，该系统由压力感应器、压力监视器、压力控制器3个子构件组成：压力感应器能够响应监视器的pressure事件请求，提供压力数据；压力监视器要求至少每18ms从压力感应器采集一次压力数据，并在2ms内向控制器发出降温或升温请求；控制器在收到信息后启动降温或加热。在系统启动后，压力感应器要求至少在18ms内接到pressure事件请求，显然，压力控制器要在系

统启动 20ms 内接收到降温或加热请求。

为了方便输入，“↑”和“↓”分别用“^”和“#”表示。为方便程序处理，在 TCBV 中将事件的时间约束信息放在了事件标识的最后面。

各构件行为分别建模如下：

1)压力感应器时间行为协议(Pressure_inductor)为：

? pressure^[18];! pressure#

2)压力监视器时间行为协议(Pressure_monitor)为：

(! pressure^[18];? pressure#;(! tem_low^[20];

? tem_low#+! tem_high^[20];? tem_high#;)) *

3)压力控制器时间行为协议(Pressure_controller)为：

? tem_low^[20];! tem_low#+? tem_high^[20];

! tem_high#

将 3 个构件的时间行为协议输入 TCBV，编译后，可以看到各协议的状态迁移图，分别如图 2－图 4 所示。

图 2　压力感应器时间行为协议状态迁移图

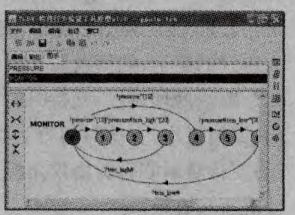

图 3　压力监视器时间行为协议状态迁移图

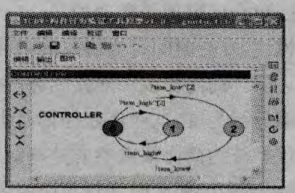

图 4　压力控制器时间行为协议状态迁移图

前两个构件组合的行为协议迁移图如图 5 所示。

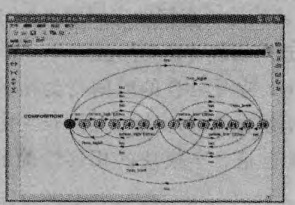

图 5　时间行为协议组合状态迁移图

因为组合协议 Composition1 会发出！tem_low[2]^，且还没有和压力控制器时间行为协议组合，故它找不到相应的响应事件？tem_high[2]^。组合结果会给出“Bad activity”提示，如图 6 所示。

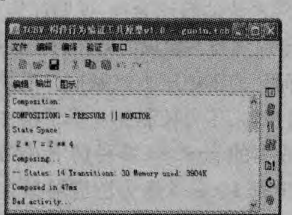

图 6　时间行为协议组合结果

3 个子构件组合成一个构件(系统/子系统)，运用 TCBV 显示组合后的图形，结果如图 7 所示。可以看出，图示已经非常复杂，因此，人工对时间行为协议组合结果进行分析是不现实的。

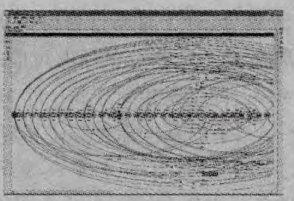

图 7　压力感应、监视和控制器组合图示

使用 TCBV 进行自动检测时，可以发现由于压力控制器时间行为协议中有事件响应 tem_low^ 事件，因此“Bad activity”错误提示消失，结果如图 8 所示。

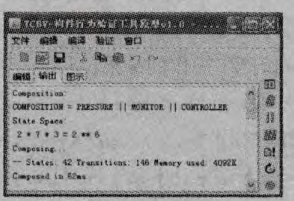

图 8　压力感应，监视和控制器组合结果

修改压力感应器时间行为协议，将接收 presure 事件的时间约束改为 10，修改后时间行为协议如下：

? pressure^[10];! pressure#

它和压力监视器组合后，结果如图 9 所示。

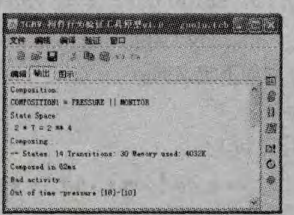

图 9　修改的压力感应和监视器组合结果

修改后的压力感应器要求在系统启动 10ms 内接收到请求事件，而压力监视器可能在 18ms 才发出请求，因此存在时限匹配错误，TCBV 提示发生“Bad activity…Out of time”错误。

4　TCBV 与其它工具的比较

我们分析了一些现有的比较有影响的模型检验工具，如 Behavior Protocol Checker(BPChecker)，LTSA，UPPAAL，SMV 等。其中，BPChecker 模型验证工具支持行为协议的形式化验证。BPChecker 是 SOFA 项目组的开源项目，该工具使用语法树自动机(Parse tree automata)来表示行为协议，组合构件的行为协议的语法树自动机可以由其子构件行为协议的语法树自动机组合而成。通过对组合而成的语法树自动机的遍历和分析，该工具能够检测构件组合中的“bad activity”、“no activity”和“divergence”错误，并能够验证两个或多个行为协议的相容性，且可以另外集成 Caesar/Aldebaran 模型检验工具进行验证。但是，该工具在状态空间搜索上采用了深度优先遍历算法(DFS)，其生成的反例过长，不利于分析解释和错误的查找。另外，该工具使用命令行方式操作，不利于应用推广，不能够直观形象地给出行为协议的图形化表示形式。

模型验证工具 LTSA 是一个基于进程代数 FSP 的验证

工具，支持对 FSP 进程的分析、图形化显示和组装，并可以对进程进行安全性、活性等性质的检测。但 LTSA 对系统状态采用显式(explicit)表示，验证中占用的存储空间过大。FSP 不支持对时间系统的建模。TCBV 使用并改进了 LTSA 的一些模块如协议编辑、图形化显示等部分来支持对 TBP 的编辑、相容性(安全、活性)的验证，增加了时间行为协议约简、时间属性的描述和可替换性验证等功能。与 LTSA 相比，TCBV 在时间行为协议的状态约简、实时性质验证等方面功能更强大。

TCBV 的设计注重功能完善、简单易用，系统的用户界面简明清晰，窗口功能以及菜单、按钮选项意义明确，力图通过简单的文本输入、编辑和单击操作，就可在工具视图中进行时间行为协议描述和验证；并注重健壮可靠，支持时间行为协议的语法分析，对语法错误在原文本上给出提示，对于用户所有可能的错误输入，能够给出出错信息提示。

结束语 各种复杂 CPS 构件系统在航空航天、医疗、交通等各种高可靠需求领域的应用越来越广泛，对复杂实时构件系统行为进行形式化描述与验证可以提高系统的正确性、可靠性。本文介绍了基于时间行为协议的时序行为形式化建模与组合相容性验证方法，给出了建模与验证工具 TCBV 的框架与功能模块，结合验证工具给出了应用实例。结果表明，基于时间行为协议的建模和验证方法可以对复杂实时构件系统时序行为进行准确的建模，并可以方便地检验系统的各种时序行为错误。在今后，我们将进一步开展更复杂的现实应用系统时序行为的建模、验证工作。

参 考 文 献

[1] 李建中. 信息物理融合系统(CPS)的概念、特点、挑战和研究进展[R]. 2009 年中国计算机科学技术发展报告，2010：1-17

[2] 贾仰理，李舟军，邢建英，等. 基于模型检验的构件验证技术研究进展[J]. 计算机研究与发展，2011，48(6)：913-922

[3] 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)：143-1147

[4] Plasil F, Visnovsky S. Behavior Protocols for Software Components[J]. IEEE Transactions on Software Engineering, 2002, 28(11): 1056-1076

[5] He Ji-feng. A Classical Mind: Essays in Honour of C. A. R. Hoare[M]// Roscoe A W, ed. International Series in Computer Science. Prentice Hall, 1994: 171-189

[6] 郭亮，唐稚松. 基于 XYZ/E 描述和验证容错系统[J]. 软件学报，2002，13(5)：913-920

[7] Lynch N, Segala R, Vaandrager F. Hybrid I/O automata[J]. Information and Computation, 2003, 185(1): 105-157

(上接第 138 页)

现后对软件系统的可靠性进行评估，并不适用于软件设计早期对设计方案进行预测和评估。

基于累加的评估方法通过系统测试获得各组件的执行时间和相对使用率，计算单个组件的失效数和失效率函数，该方法并不直接考虑软件系统的构架，而是间接通过组件的可靠性来估计系统架构，如 M.Xie 提出的附加可靠性模型[13]。此类方法同样是在软件实现后对软件系统的可靠性进行评估。

运用蒙特卡洛方法时无需考虑复杂的系统结构，因此它特别适合大规范复杂的面向服务软件系统的建模与分析。本文在运用该方法时，提出使用着色 Petri 网的形式化描述方法取代可靠性方框图。一方面，着色 Petri 网可以清晰地描述面向服务软件中包括异步并发在内的各种复杂结构。另一方面，着色 Petri 网有成熟的工具支撑，可以对各模块的故障情况进行模拟仿真。

结束语 本文利用着色 Petri 网作为面向服务软件异常处理逻辑的形式化描述工具，并借鉴传统的组件重要性评估方法，提出一种面向服务软件中异常处理模块重要性的仿真分析方法，该方法可以为软件优化提供定量分析依据；接着对传统软件可靠性评估方法的特点及其不足进行了分析。

蒙特卡洛方法的采样次数与系统规模无关，它容易处理各种实际运行控制策略，故可以用于大型面向服务软件系统的可靠评估。但蒙特卡洛方法存在计算时间与计算精度的矛盾，即为了获得精度较高的可靠性指标，需要进行长时间的模拟计算。如何利用方差加快蒙特卡洛模拟的收敛速度，提高仿真效率，可以作为今后的研究方向。

参 考 文 献

[1] Carbone M. Session-based Choreography with Exceptions[J]. Electronic Notes in Theoretical Computer Science, 2009, 241(C): 35-55

[2] Diez F J, Maurtua I. Dynamic Exception Handling Based on Web Services and OPC XML-DA[C]// Proceeding of IEEE International Conference on Web Services (ICWS08). IEEE Computer Society, 2008: 593-599

[3] Men Peng, Duan Zhen-hua, Yu Bin. Utilizing Fuzzy Petri Net for Choreography Based Semantic Web Services Discovery[C]// ICATPN 2007. 2007: 362-380

[4] Gokhale S. Quantifying the variance in application reliability[C]// Proceedings of the Pacific Rim Dependability Conference. Papeete, Polynesia, 2004: 113-121

[5] Siegrist K. Reliability of systems with Markov transfer of control[J]. IEEE Transactions on Reliability, 1988, 14(7): 1049-1053

[6] 陆文，徐锋，吕建. 一种开放环境下的软件可靠性评估方法[J]. 计算机学报，2010，33(3)：452-462

[7] Cristian F. Exception handling and tolerance of software faults [J]. IEEE Transactions on Computers, 1982, 31(6): 531-540

[8] 柳永坡，吴际，金茂忠. 基于贝叶斯统计推理的故障定位实验研究[J]. 计算机研究与发展，2010(4)：707-715

[9] Siegrist K. Reliability of systems with Markov transfer of control[J]. IEEE Transactions on Reliability, 1988, 14(7): 1049-1053

[10] 刘延夫. 基于马尔可夫分析方法的软件系统可靠性研究[D]. 长春：长春理工大学，2006

[11] Littlewood B. Software reliability model for modular program structure [J]. IEEE Transactions Reliability, 1979, 28(3): 241-246

[12] Sherif M. Scenario-based reliability analysis of component-based software[C]// 10th International Symposium on Software Reliability. Boca Raton, Florida, 1999: 22-31

[13] Xie M, Wohlin C. An additive reliability model for the analysis of modular software failure data[C]// Proceedings of the Sixth International Symposium on Software Reliability Engineering (ISSRE'95). 1995: 188-194