一种基于规范的可信协同系统分析设计框架

黄 勇[1] 潘雪增[1] 蔡国永[1,2] 平玲娣[1]

(浙江大学计算机科学与技术学院 杭州 310027)[1] (桂林电子科技大学计算机与控制学院 桂林 541004)[2]

摘 要 针对可信协同系统分析建模与设计问题,提出了RBN-T模型,分析了模型中的可信保障机制及策略。提出了基于RBN-T模型的可信协同系统分析建模过程,以具体的实例说明了RBN-T模型在可信协同系统分析建模与设计上的可用性。RBN-T模型把基于角色的规范管理提升到适合可信协同系统分析建模与设计的层次,从而有利于在协同系统开发的早期阶段就关注可信问题。

关键词 可信系统,协同系统,角色,规范,需求建模

中图法分类号 TP393 **文献标识码** A

Norm Based Framework toward Modeling and Designing of Trusted Collaborative Systems

HUANG Yong[1] PAN Xue-zeng[1] CAI Guo-yong[1,2] PING Ling-di[1]

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)[1]

(College of Computer and Control, Guilin University of Electronic Science and Technology, Guilin 541004, China)[2]

Abstract To deal with the issue of modeling and designing trusted collaborative systems, a RBN-T model was proposed, and the trust guarantee mechanisms and their policies were also discussed. To illustrate the usability of the RBN-T model, a modeling procedure based on RBN-T was given and a case study was presented. RBN-T raises the normative role based management approach toward a higher level which facilitates the concerns of trust into the early phases while developing trusted collaborative systems.

Keywords Trusted system, Collaborative system, Role, Norm, Requirement modeling

在开放的网络支撑环境(如因特网)上,如何开发可以信赖的协同工作系统是当前的一个热点研究问题[1,2]。对此已开展了许多卓有成效的工作,达成了一些基本的共识。如Haibin[3,4]系统阐述了角色概念在协同系统中的重要作用,向勇[2]系统地分析了协同工作系统研究现状,认为协同工作系统是一个综合交叉的领域,除了计算机学科外,还需要更多借助社会、心理和经济等学科的基本概念和原理。尽管协同系统研究取得了很多的成果,但要得到更广泛的实际应用,还有许多重要的方面需要进一步研究解决。如在开放的环境中,协同的自治主体存在各自的兴趣和利益,因此如何保证安全可靠并成功地实现协同任务的完成就是一个十分重要的问题(本文称之为可信问题)。传统的协同软件开发方法对此还没有加以充分的重视,特别是没有把可信方面的关注点融入协同系统开发的分析建模过程中。

与可信问题紧密相关的一个方面是访问控制。目前在访问控制模型方面已取得了很多重要的成果,最重要的是提出了具有充分柔性的RBAC模型[5]。借助RBAC模型的方法可以提高协同系统的安全性。然而由于协同主体的自治性,主体的动作并不能完全受控制,协同任务又需要主体的协作完成,因此RBAC模型并不足以解决可信问题。本文认为在RBAC模型基础上,扩充更多社会学与经济上的概念,如义务、奖罚或激励等,将能更有效地促使自治主体完成协作任务,从而提高协同系统的可信性。

本文在第1节中提出一个可信协同模型RBN-T;第2节对RBN-T模型及其可信机制展开讨论;第3节提出基于RBN-T模型的可信协同系统设计过程;第4节介绍基于RBN-T模型的实用案例;第5节讨论一些相关工作;最后总结本文。

1 可信协同管理模型RBN-T

社会经济学理论认为,自治的主体尽管不完全受控,但它存在一定的利益追求,因此如果其参与的协同工作能够促使其利益得到实现,则它将很可能会履行其协同工作中的职责或义务,从而使得协同任务得到成功完成。本文提出在RBAC模型中引入义务和政策元素来表达对协同主体行为的管理,得到称之为RBN(Role Based Norm)的模型。

定义1 RBN模型是一个多元组 $RBN=\langle U,S,R,M,N,L,F\rangle$,其中 U,S,R 同RBAC模型,分别表示用户集、会话集、角色集,$M\in R$ 为一个特殊的角色,称管理角色;N 称为角色规范,$N=\langle A,O,P,D\rangle$,分别表示授权集 A、义务集 O、政策

到稿日期:2008-05-05 本文受国家863高技术研究发展计划资助项目(2006AA01Z431),浙江省重大科技专项重点项目(2007C11068&2007C11088)资助。

黄 勇 博士生,研究方向为信息安全和可信软件开发等,E-mail:hy970531@sina.com;潘雪增 教授,博导,研究方向为信息安全等;蔡国永 博士,副教授,研究方向为软件工程等;平玲娣 教授,博导,研究方向为信息安全等。

集 P 和资源集 D;L 表示 RBN 模型元素上定义的关联集,$L=\langle UA,RA,RO,RP\rangle$,分别表示用户与角色关联集 $UA\subset U\times R$,角色与授权关联集 $RA\subset R\times A$,角色与义务关联集 $RO\subset R\times O$,角色与政策关联集 $RP\subset R\times P$;F 表示 RBN 上定义的映射函数集。

核心映射函数集通常在关联集上定义,类似 RBAC 的做法,如在关联集 $UA\subset U\times R$ 可以定义映射函数:$assigned\_users(r)=\{u\in U|(u,r)\in UA\}$。其它扩展的一些映射函数集往往与具体的实现有关,故本文不对此进行详细讨论。相对于 RBAC 模型,RBN 模型除了增加了一个义务元素和一个政策元素,以及角色的义务关联和政策关联外,同时对角色的权限、义务和政策解释也赋予了一些新的含义。在 RBN 模型中,角色表示协同工作的用户主体的管理抽象,角色义务表示角色在协同工作组织中要承担的职责。在 RBN 中,角色权利表示角色主体为完成其职责可以使用的资源,政策表示对角色完成其职责的处置策略,特别是激励处罚政策,如角色主体履行其职责,他将会得到奖励,反之则会受到处罚。当然也可以表达其它系统需要的异常处理策略。

尽管 RBN 模型提供了丰富的协同规范管理语义,但 RBN 模型是一个抽象的、独立于业务的管理模型。在实际协同系统的开发中,角色的权限义务及处置政策是与协同系统要完成的工作任务有关的。因此要把 RBN 应用到协同系统的分析建模阶段,RBN 模型就必须与协同工作的系统任务模型相融合。基于此,把 RBN 模型进一步扩展到称为 RBN-T 的模型。

定义 2 RBN-T 模型表示为一个元组 $RBN\text{-}T=\langle U,S,R,M,N,L,F,T,C\rangle$,其中 U,S,R,M,N 与 RBN 模型相同,T 表示协同任务集,C 表示上下文环境,L 表示上述元素间的关联。$L=\langle UA,RA,RO,RP,TA\rangle$,其中 $TA\subset T\times R$ 表示任务到角色的分配,其它与 RBN 中的关联集相同。

RBN-T 模型可图形化表示为一个三角形结构。三角形的三个顶点(虚线矩形)分别表示协同用户、协同任务、协同资源。三条虚线边分别表示三个顶点元素之间可能存在的联系。在三角形的内部是 RBN 模型,通过 RBN 模型,协同用户、协同资源、协同任务间的物理关联才能实际建立。在三角色的外部是协同系统的上下文环境(context),用于建立表示用户、资源和任务的环境度量属性,如图 1 所示。

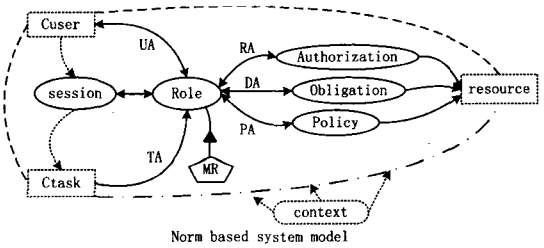

图 1 RBN-T 模型

图中的实线箭头表示关联关系的度,如协同主体与角色之间是多对多的关系(用双向箭头线表示),即一个主体可以扮演多个角色,一个角色也可以由多个主体扮演。从用户主体到会话,再到任务的单向虚箭头线表示主体启动会话,企图执行某个任务。角色在执行任务的过程中就获得了一定的权限,具有了一定的义务,其执行情况受相应的处置政策的监管。如此扩充之后,协同任务模型就整合到了 RBN 模型中,形成了在需求分析建模阶段可以采用的 RBN-T 模型。

2 RBN-T 模型分析

RBN-T 可以看作是一个面向规范的多主体可信协同系统模型,为了能更好地理解这一模型,并应用到协同系统分析设计过程中,下面对 RBN-T 模型进行进一步的讨论,特别是与可信协同保障相关的机制及策略问题。

2.1 RBN-T 模型的规范焦点

从规范的角度看,在 RBN-T 模型中包含 3 个规范子模型:访问授权规范(RBA),义务分配规范(RBO),处置政策规范(RBP)。每一个规范子模型都以角色这个重要概念为纽带把规范焦点联系到用户和任务,同时角色这个概念也把 3 个规范子模型整合到一个框架之中。这 3 个规范子模型可以重新组合,形成新的子模型,如授权与义务结合的子模型、授权与政策结合的子模型、义务与政策结合的子模型。从而这些子模型间形成如图 2 所示的分解组合关系。其中虚箭头线表示分解关系,即 RBN-T=RA-T+RP-T+RO-T;实箭头线表示组合关系,即 RA-T+RO-T=RAO-T,RA-T+RP-T=RAP-T,RO-T+RP-T=ROP-T。

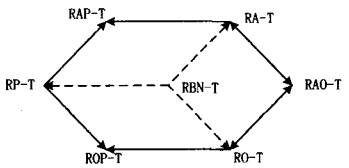

图 2 RBN-T 模型的分解组合结构图

2.2 RBN-T 中的可信保障机制及策略

RBAC 主要关注授权的安全性,因此主要采用了两种机制:分离机制和检查机制,如静态职责分离、动态职责分离和关联度约束检查等。在 RBN-T 模型中,为了保障协同任务的可信完成,除了保持分离机制外,对检查机制进行了扩充,另外引入两种新的机制:激励处罚机制、重构机制。

在 RBN-T 模型中,检查机制分静态语义和动态语义检查。静态语义检查主要指一致性检查。如某角色有某一义务,而完成该义务需要的资源访问并没有被授权,则义务与权限之间将存在不一致。静态语义检查可以通过语法分析进行。而动态语义检查则只能通过在线监视完成。本文主要针对静态语义检查进行分析。

静态语义不一致性也称模态冲突,主要的规范模态冲突类型可分为权限、义务、权限义务等几种模态冲突[5]。为便于形式化定义这些冲突,我们采用道义逻辑(Deontic logic)[6]公式描述并做了相应的扩展。其中,$permit(\varphi\leqslant d|\rho)$,$forbid(\varphi\leqslant d|\rho)$,$obligate(\varphi\leqslant d|\rho)$分别表示在时限 d 到达之前,如果条件 ρ 成立,则有义务、允许或禁止实施 φ。此外,另外引入 $reward(\varphi\leqslant d|\rho)$和 $punish(\varphi\leqslant d|\rho)$表示在时限 d 到达之前,如果条件 ρ 成立,则给予信誉 φ 奖励或处罚。

类型 1(权限模态冲突) 指权限表达时,如果同时使用否定授权模态语句和肯定授权模态语句,则可能同时存在一个允许的授权说明语句和一个禁止的授权说明语句。令 $n=permit(\varphi\leqslant d|\rho)$,$n'=forbid(\varphi'\leqslant d'|\rho')$为两条权限规范,如果 $\varphi\cap\varphi'\neq\phi\wedge d\cap d'\neq\phi\wedge\rho\cap\rho'\neq\phi$,则规范 n 和规范 n' 存在

权限模态冲突。

类型2(义务模态冲突) 指义务规范的目标动作或状态公式之间存在冲突或矛盾,从而导致义务的不确定性。令 $n=obligate(\varphi\leqslant d|\rho)$,$n'=obligate(\varphi'\leqslant d'|\rho')$表示两条义务规范,如果 $\varphi\cap\varphi'=\phi\wedge d\cap d'\neq\phi\wedge\rho\cap\rho'\neq\phi$,则称规范 n 和规范 n'存在义务模态冲突。

类型3(权限义务模态冲突) 指对同样的操作或状态公式,可能存在授权与义务说明的不一致性。如可能存在这样的模型描述:有义务但无权利或有权利但无义务。令 $n=obligate(\varphi\leqslant d|\rho)$为一条义务规范,$n'=forbid(\varphi'\leqslant d'|\rho')$为一条授权规范,如果 $\varphi\cap\varphi'\neq\phi\wedge d\cap d'\neq\phi\wedge\rho\cap\rho'\neq\phi$,则称规范 n 和规范 n'存在权限义务模态冲突。

类型4(奖励处罚模态冲突) 指对同样的操作或状态公式,可能存在奖励与处罚的不一致描述。令 $n=reward(\varphi\leqslant d|\rho)$为一条奖励规范,$n'=punish(\varphi'\leqslant d'|\rho')$为一条处罚规范,如果 $\varphi\cap\varphi'\neq\phi\wedge d\cap d'\neq\phi\wedge\rho\cap\rho'\neq\phi$,则称规范 n 和规范 n'存在奖励处罚模态冲突。

定义3(RBN-T模型规范一致性) RBN-T模型规范如果不存在权限模态冲突、义务模态冲突、权限义务模态冲突和奖励处罚模态冲突,则称RBN-T模型规范一致性。

义务能否按时完成是协同任务能否成功实现的重要方面,这是动态语义检查的一个重要方面。除了保证业务流程本身的活性外,协作主体的执行意愿就是一个重要的因素,为激发协作主体的意愿,RBN-T模型中引入了激励处罚机制和重构机制。激励处罚主要针对协同主体利益或价值追求设计,可以引入信誉或货币作为度量参数。根据主体履行义务的质量概率来测量,如履行质量高,则奖励主体一定的货币量,否则收回其部分货币量。当主体的货币数量超出一定的阈值范围时,则启动系统的重构机制,如重新调整承担角色的主体,或调整主体承担的角色数量等。系统通过重构进化,将最终达到一个高可信度的配置,保证协同任务的成功完成。

3 基于RBN-T的系统分析设计过程

RBN-T模型为协同系统建立了一个抽象的元模型,提供了协同系统工作的基本语义描述空间。然而基于RBN-T元模型进行协同系统的分析设计,还需遵循一定的过程。具体而言,基于RBN-T的分析设计遵循以下几个步骤:

(1)领域任务分析:理解协同的领域任务,对任务进行分解,直到不再需要分解的协作原子任务为止。

(2)领域角色分析:根据领域任务和实际环境的需要,识别协同系统的用户类型并标识为主角色,然后根据任务的分解树,对主角色进行分解或直接分配到相应的原子协同任务。同时标识角色的义务。

(3)领域任务资源分析:分析任务树中的每一个原子任务的执行需要的信息或数据资源,以及对该资源涉及的操作,同时标识出对应角色或子角色的权限。

(4)基于角色建立领域协同任务的交互模型:根据协同任务或子任务,分析上述标识出的协同角色和资源,建立它们之间的交互过程模型。

(5)通过分析角色交互过程模型,进一步标识出角色的权限和义务及检查其一致性。

(6)对角色的权限和义务,针对4种可信保障机制,分别设计相应的分离策略、检查策略、激励处罚策略和重构策略。

(7)重复上述过程,直到得到满意的分析结果。

(8)根据上面分析的结果,进行协同系统体系结构的选择和总体设计。

3.1 领域任务角色分解模型

为了描述协同任务的细化,在传统任务分解树结构的基础上,本节提出下面的角色任务分解模型,它包括两个层面,一是任务的分解,二是原子任务的角色及资源的分配。

定义4 角色任务分解模型表示为 $TRS=\langle T,H,R,S\rangle$,其中 T 表示任务集,H 表示任务之间的分解关系集,R 表示原子任务涉及的角色集,S 表示原子任务涉及的资源集。对于不再分解的原子任务,表示为$\langle t,\bigcup_{i=0..k} r_i,\bigcup_{j=0..n} s_j\rangle$,其中 $t\in T,r_i\in R,s_j\in S$。

不同于一般的任务分解树,在这里任务分解树中的原子任务分成两大类:非协同原子任务和协同原子任务,非协同原子任务指该原子任务需且仅需一个角色参与;而协同原子任务指需要两个以上参与角色的原子任务。由两个角色参与的协同原子任务,通常也称二元协同原子任务,由两个以上角色参与的协同原子任务通常也称多元协同原子任务。对于复杂的多元协同可以进一步分解,使之成为多个二元协同的复合。不管是否是协作原子任务,通常都涉及一定的资源操作。

3.2 角色交互与规范策略

为进一步明确角色之间的交互过程并考察其中的权限、义务及处置政策,在RBN-T模型中提出一个基于规范的UML顺序图的扩展,称为角色规范交互顺序(RNS)图。RNS中的交互对象有角色和资源,交互的传递消息分成4类:自由请求消息、响应消息、授权请求消息、义务请求消息。形式地,$RNS=\langle R,S,I\rangle$,其中 R 表示角色集,S 表示资源集,I 表示交互传递消息集。

图3示例了一个有两个角色、一个资源的二元交互顺序图,交互消息按从上往下的顺序排列,实箭头线表示自由请求消息,带起始圆点的实箭头线表示资源请求的响应消息,自由请求和资源的响应消息都不需要规范显式管理的消息。虚箭头线表示义务请求消息类型,双箭头线表示授权请求消息类型。

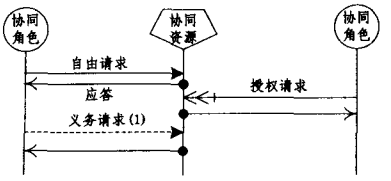

图3 角色规范交互顺序图

RNS模型和TRS模型一起为RBN-T模型的建立提供了基础,在RNS和TRS的支持下,RBN-T的主要任务就是进一步建立授权、义务和政策,并采用第2节中的方法分析规范的一致性。

4 RBN-T模型的应用

本节以我们实现的出版公司稿件协同处理系统(PHS)为案例来进一步说明RBN-T模型在可信协同系统设计中的可用性。在稿件协同处理系统中涉及的协同任务是:稿件的收集、出版和营销。其基本过程是:出版公司根据市场的情况,确定某一有市场前景的主题征集稿件,经过评审专家评审后,选择整理出版相关的电子图书专辑,并进行市场化营销推广。

4.1 PHS系统任务角色分解模型

经过一定的简化处理后，PHS 系统的基本任务分解为五个，分别为：稿件征集、稿件评审、稿件出版、稿件营销、账务管理。涉及的主要角色有：作者、评审员、总编、出版编辑、销售员、财务员、管理员。假设稿件征集又分解成两个原子任务：征稿信息发布、接收投递稿件；稿件评审分解成稿件分配和接收评审意见。则部分 PHS 的任务角色分解模型如图 4 所示。

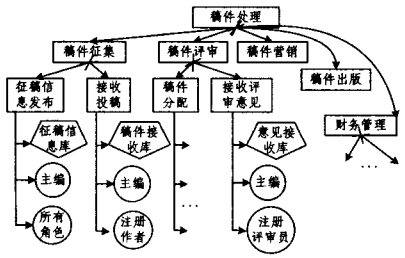

图 4　PHS 系统的任务角色分解模型

4.2　PHS 系统的原子任务角色交互模型

对于 PHS 中的原子任务，可以进一步建立其角色交互过程模型。以稿件分配与评阅交互为例，可以建立图 5 所示的 RNS 图。其它原子任务角色交互模型可类似建立。

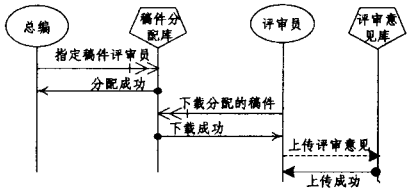

图 5　PHS 系统的部分角色规范交互图

4.3　PHS 的角色规范

经过上述的分析后，可以建立角色的权限、义务和政策规范，PHS 系统的部分角色规范如表 1 所列。

表 1　PHS 系统的角色规范

角色	权限	义务	政策	备注
作者	查询征稿信息、查询所投稿件的处理信息(评审、出版、稿酬、自身账号信息)	撰写稿件、投稿	如果长期不投稿，则通知取消其账号；如果投稿录用率高，则升级为荣誉作者。	
总编	查询稿件信息库、评审意见库、稿件出版、销售、账务信息	指定稿件的评审员，决定录用的稿件，设置征稿信息	如果不能按时指定稿件评审员，其信誉货币将会降低，同时将会对其提供提醒。不能把评审员自己的稿件分配给评审员评审。……	
评审员	查询征稿信息、查询评审报酬	下载评审稿件、按时提交稿件评审意见	如果所评审稿件被录用出版的比率高，则升级为资深评审员；如果不按时提交评审意见，则降低评审信誉；反之，则提高其评审信誉。如果信誉值超过一定范围，则停止其评审员资格，重新招募新的评审员。	
…	…	…	…	

5　相关工作

基于角色和规范的思想实现协同系统的管理已在许多研究文献上提出，并被认为是一个重要的进步。文献[1]对互联网软件的可信机理进行了探讨，提出了一个基于反馈的可信保障机制，但在软件开发时如何应用该模型尚未讨论。文献[3]讨论了基于角色的协同系统的基本问题，并提出一个基于角色协同的基本模型 E-CARGO[4]，虽然 E-CARGO 模型基于角色，但并没有从规范的角度考虑，也未讨论可信协同问题。文献[7]提出了需要从高层抽象开始设计安全系统模型，并提出从组织的任务、授权、职责等组织外部接口需求开始，进而细化到组织内部的需求来进行建模，并提出一些符号来表示组织应用模型，但文献中并没有给出具体可用的模型。文献[8]也对基于角色的协作关系建模进行了讨论，给出一些建模原则及表示符号，但没有给出协同系统的模型，也没有讨论协同关系中的规范处理问题。文献[9]从需求工程的角度，讨论如何用角色来建立访问策略的方法，文献[10]提出从规范的角度来考虑协同系统的设计，讨论了规范表示的规则式语言，并对一些规范规则进行了表示。文献[11]针对 agent 协同的问题，提出了对基于目标与面向角色分析方法的改进。本文提出的 RBN-T 方法的思想与文献[7-11]有一定类似，但做法是完全不同的。我们特别针对协同系统的可信问题，从 RBAC 模型开始，逐步扩充规范概念，融入协同任务概念，最后建立了完整的可信协同系统模型，并提出了基于 RBN-T 模型的具体开发过程与其它一些技术，使之成为一个更具体适用性的开发框架。我们的方法可以看作是对上述研究工作的综合补充与完善。

结束语　协同工作系统是一类有广泛应用前景的重要的信息系统形式，保证协同系统的可信是一个重要的方面。从 RBAC 模型出发，本文提出了一个基于角色规范的可信协同系统模型 RBN-T，分析了 RBN-T 中的可信保障机制及规范策略，然后提出了基于 RBN-T 模型的可信协同系统开发的过程框架，最后以所实现的实用案例说明了 RBN-T 的应用。尽管 RBN-T 为可信协同系统的开发提供了一种较好的分析建模的途径，然而在基于 RBN-T 方法的开发中还有一些技术问题需要开展进一步的研究，如如何用形式化的方法来描述 RBN-T 模型，以便于模型的自动分析与管理。

参 考 文 献

[1]　王怀民，唐扬斌，尹刚，等. 互联网软件的可信机理[J]. 中国科学(E 辑)，2006，36(10)：1156 -1169

[2]　向勇，张少华，史美林. 国内协作研究的现状和发展[J]. 通信学报，2006，27(11)：1-6

[3]　Zhu Haibin. Some Issues of Role-based Collaboration[C]//Canadian Conference on Electrical and Computer Engineering，2003

[4]　Zhu Haibin，Zhou MengChu. Role-Based Collaboration and Its Kernel Mechanisms [J]. IEEE Transactions on systems，man and cybernetics，2006，36(4)：578-589

[5]　蔡国永，林煜明. RBAC 模型的扩充及其应用[J]. 计算机机工程与应用，2008，44(3)：228-233

[6]　Broersen J，Dignum F，Dignum V，et al. Designing a Deontic Logic of Deadlines[C]// Proceedings of the 7th International Workshop on Deontic Logic in Computer Science. 2004

（下转第 165 页）

安全性质偏重于判断系统未来是否在某种情况下达到某个状态，而活性性质偏重于验证未来任何情况下都能达到某个状态。表示这些性质的公式的真与假不是固定不变的。时态逻辑可以用来表示这些待验证的性质。时态逻辑包含若干状态，公式的真假随着状态变化而变化。线性时间逻辑把未来看成状态序列，逻辑公式的满足性问题需要判断从某个状态出发的所有序列。而分支时间逻辑把未来看成分支树，能够表达一条序列是否存在的断言。

2.2 实验验证

为了验证 BVM 的可行性，本文进行了下列实验以及结果分析。图 3 是含有补偿行为的一个 BPEL 的 BVM。实验中使用的模型检测器是 NuSMV[8]。NuSMV 以描述系统的模型和时态逻辑组成的代码作为输入。若性质被满足，它产生输出结果"真"，否则产生一个不被满足的反例。实验中的模型描述了补偿行为从开始执行到结束的过程。

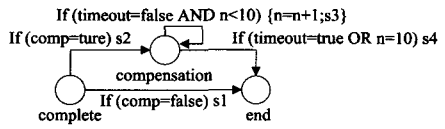

图 3 实验中的 BVM

实验的第一步是用 NuSMV 规定的语言来表达 BVM 和相关性质。部分代码如下所示，主要定义了模型的基本结构和待验证的性质。

```
MODULE main
VAR
state:{complete,compensation,end };
n:0..10;
timeout:Boolean;
comp:Boolean;
behavior:{s1,s2,s3,s4};
ASSIGN
init(state):= complete;
init(n):= 0;
init(behavior):=s1;
SPEC
AG(state=compensation->EF(state=end))
```

模型中含有 3 个状态 complete，compensation 和 end，共同组成 BVM 中的 S。$timeout, comp \in V_p$，分别表示网络等待超时和是否存在补偿行为。$n \in V_B$，表示补偿调用者的计数器(实验中假定上限值为 10)。

实验的第二步是运行 NuSMV。结果显示模型满足图 3 中的性质。如果更换将要检测的性质为 AG (state = complete -> EF (state=compensation))，则 NuSMV 将返回以下信息：

```
-- specification AG (state = complete -> EF state = compensation) is false
-- as demonstrated by the following execution sequence
Trace Description:CTL Counterexample
Trace Type:Counterexample
-> State:1.1 <-
  state = complete
  n = 0
  timeout = 0
  comp = 0
  behavior = s1
```

从以上实验可以看出，BVM 在 NuSMV 中能够检测出性质是否满足，并能给出不满足时存在的反例。

结束语 本文讨论了将 BPEL 应用程序转换为 BVM 的过程，对下一步的验证过程做了初步讨论。BPEL 应用程序的结构复杂，需要一种途径来理清逻辑框架。此外，BPEL 主要面向需求领域的业务专家，不能很好地应用到计算机自动化验证方面。BVM 在以上两个方面作了有益的尝试。将来的工作是将该 BVM 自动转换为模型检测器的输入语言(描述待验证系统的语言)，再运行检测器验证是否满足特定的性质。在此过程中，状态爆炸问题可能使得验证过程不能结束，于是有待于对 BVM 进一步优化。

参 考 文 献

[1] Curbera F, et al. Business Process Execution Language for Web Services. Version 1.1. BEA, IBM, Microsoft, SAP AG and Siebel Systems. May 2003

[2] Clarke E, Grumberg O, Peled D. Model Checking. MIT Press, December 1999

[3] Koehler J, Tirenni G, Kumaran S. From business process model to consistent implementation: A case for formal verification methods// Proc. 6th IEEE International Enterprise Distributed Object Computing Conference (EDOC). 2002:96-106

[4] Nakajima S. Model-checking Behavioral Specification of BPEL Applications. Electronic Notes in Theoretical Computer Science, 2006,151(2):89-105

[5] Zheng Y, Krause P. An automatic test generation framework for BPEL web services. Technical Report. England: University of Surrey, 2006

[6] Huth M, Ryan M. 面向计算机科学的数理逻辑：系统建模与推理[M]. 第二版. 北京：机械工业出版社，Spinger，2007：115

[7] Sharygina N, Kröning D. Model Checking with Abstraction for Web Services. Test and Analysis of Web Services, Spinger, 2007:121-145

[8] Cimatti A, Clarke E, Giunchiglia F, et al. NUSMV: A New Symbolic Model Verifier // Proceedings Eleventh Conference on Computer-Aided Verification (CAV'99), number 1633 in LNCS. Springer, 1999:495-499

(上接第 154 页)

[7] Roshan K, Sandhu R. Conceptual Foundations for a Model of Task-based Authorizations [C] // IEEE Computer Security Foundations Workshop. 1994

[8] 葛声，孙瑛霖，杜宗霞. 基于角色的协作关系建模研究[J]. 计算机机工程与应用，2003(3)：14-19

[9] Crook R, Ince D, Nuseibeh B. Modelling access policies using roles in requirements engineering[J]. Information and software technology, 2003(45):979-991

[10] Liu Kecheng, Sun Lily, Dix A, et al. Norm Based Agency for Designing Collaborative Systems[J]. Information Systems Journal, 2001(11):229-247

[11] Kuan Peipei, Rarunasekera S, Sterling L. Improving Goal and Role Oriented Analysis for Agent Based Systems[C]//Proceedings of the 2005 Australian software engineering conference. 2005