面向企业级流程的职责分离框架及其冗余分析

翟治年[1],卢亚辉[2],奚建清[3],赵铁柱[4],汤德佑[3],顾春华[5]
(1.浙江科技学院 信息与电子工程学院,浙江杭州 310023;2.深圳大学 计算机与软件学院,广东深圳 518060;3.华南理工大学 计算机科学与工程学院,广东广州 510006;4.东莞理工学院 工程技术研究院,广东东莞 523808;5.华东理工大学 信息科学与工程学院,上海 200237)

摘 要: 为企业级工作流授权机制定义了多维可泛化的职责分离框架,能够对团队任务涉及的多种分工形式进行深入全面的限制.系统分析了框架中的约束覆盖规则,并证明其正确性和完备性,为约束管理自动化奠定了基础.作为应用,根据规则给出了冗余动态约束的检测算法.最后通过案例研究验证了模型特性.

关键词: 访问控制;任务;角色;细粒度职责分离;冗余约束;
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2013) 10-2087-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2013.10.033

Enterprise-Level Business Process Oriented Framework for Separation of Duties with Its Redundancy Analysis

ZHAI Zhi-nian[1], LU Ya-hui[2], XI Jian-qing[3], ZHAO Tie-zhu[4], TANG De-you[3], GU Chun-hua[5]
(1. *School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou, Zhejiang* 310023, *China*; 2. *School of Computer and Software, Shenzhen University, Shenzhen, Guangdong* 518060, *China*; 3. *School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong* 510006, *China*; 4. *Engineering and Technology Institute, Dongguan University of Technology, Dongguan, Guangdong* 523808, *China*; 5. *School of Computer and Software, East China University of Science and Technology, Shanghai* 200237, *China*)

Abstract: Based on some enterprise-level workflow authorization mechanisms, a multi-dimensional and generalizable framework for Separation of Duty is specified, and multiple labor dividing forms related to team-collaborated tasks can be restricted deeply and all-sidedly. Coverage rules among these constraints are analyzed systematically. The correctness and completeness of these rules are proved such that a basis for the automation of constraint administration is provided. As application of the rules, a detecting algorithm for redundant dynamic constraints is given. Finally, the features of this model are verified via a case study.

Key words: access control; task; role; fine-grained separation of duties; redundant constraint

1 引言

作为企业级协同应用,业务过程管理涉及大量主客体对象,必须以可伸缩易维护的方式进行权限管理.2003 年,Oh 等建立了权限-任务-角色-用户三步授权机制[1],通过角色及其管理层次简化了任务分配.2011 年,翟等提出基于任务泛化的继承方式,解决了该机制中任务间重复授权的问题[2].三步授权机制以 All-or-Nothing 的方式将任务(及其权限)分配给角色,不支持多个角色以不同权限协作完成任务.2012 年,翟等在基于任务-角色指派关系的授权机制[3]中引入关联继承机制,可兼顾授权伸缩性和团队任务支持[4],但未讨论其约束定义和管理问题.团队任务涉及的多种分工形式均须进行职责分离(Separation of Duties, SoD),特别是复合职责要求细粒度的 SoD.每种分工形式自身有不同粒度,相应的 SoD 应当可以泛化,以免在细小粒度上重复表达类似的约束.

面向业务过程的约束研究已相当丰富.1999 年,Bertino 等给出一种基于逻辑程序的约束描述语言[8],包含常/变量和多种谓词,可以对工作流定义和执行过程施加灵活的约束.2003 年,Wainer 等基于标准逻辑程序给出了类似的约束语言[7],但考虑了工作流多实例的情况.2004 年,Liu 等讨论了任务互斥的多种涵义及其对授权和激活过程的多种约束[9].2005 年,邢等给出一种使用量词和函数,并带蕴涵符的约束语言[5],可描述任务、角色和权限等多种 SoD.2010 年,Wang 等研究了抗多用户共谋的任务 SoD[6].上述约束研究均以三步授权机制或其非彻底的定义(只含任务-角色-用户关联,

收稿日期:2012-08-10;修回日期:2013-02-15;责任编辑:孙瑶
基金项目:国家自然科学基金(No.6093114,No.60973100,No.51246002,No.61272200);广东省自然科学基金(No.S2012040007746)

不考虑任务内部应用数据的访问权限,例如文献[7~9])为背景,所针对的协作模式相对简单.直接定义的约束均未将任务-角色复合职责作为约束目标,而基于逻辑程序的约束语言缺乏描述用户在某任务中担任某角色关系的谓词,它们均不能对团队任务的复合职责施加准确的约束.

另一方面,企业协同应用日益趋向跨域和开放[10,11],安全策略的数量以及规则冲突冗余的可能性激增,极大地影响了访问控制决策的合理性与效率.这就需要对策略关系进行理论分析,据此自动检测冲突和冗余,减轻人工管理负担.目前出现了不少冲突检测与消解方法[10,12],但冗余分析只有一些基于可扩展访问控制标记语言的工作[12,13].作为基于属性的规则描述框架,该语言并不特定于任何访问控制模型,相应层面上的冗余分析可得出一些宽泛的结论,但无法针对具体模型进行深入.同时,现有工作[12,13]仅给出冗余分析方法,均未讨论其完备性.

本文将对基于任务-角色指派关系授权机制的约束定义和冗余分析问题进行更为深入的研究.

2 授权机制概述

本文依赖的授权机制如图1所示.将表示工作分配关系的任务-角色二元组定义为一种复合职责,作为授权依据和用户组织单位.根据任务层次关系和角色特化关系可以导出复合职责之间的特化关系,称为关联特化.泛化复合职责的授权可被特化复合职责继承,而后者的成员也是前者的成员.任务可能有多个实例,在每个实例执行前,须为其中每个复合职责分配执行用户,称为资源分配.用户通过会话与系统进行交互.在会话中,用户首先激活自己通过资源分配关系在同一工作流案例中承担的若干复合职责,然后激活这些复合职责及其泛化复合职责的授权,最后在已激活权限的控制下访问该工作流案例中的应用数据,完成业务操作.下面分授权和访问检查两节来描述相关概念.

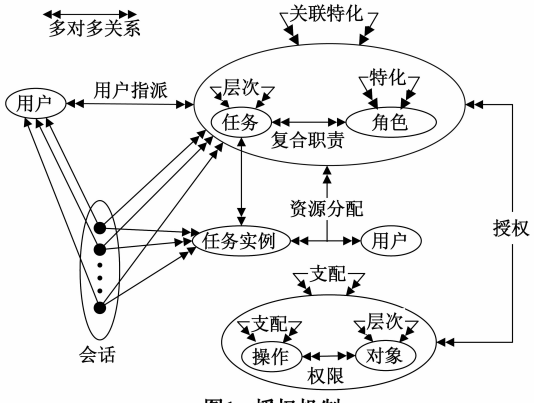

图1 授权机制

2.1 授权

本文的任务、角色、权限等等概念与文献[4]一致,这里仅给出有关的记法:任务集 T;任务层次 $TH \subseteq T \times T$;任务实例集 I,每个任务实例 i 派生于唯一的可执行任务 $t(i)$,它在时刻 t 的状态记为 $s(i,t)$,其值为1表示正在执行,为0表示未开始或已结束;角色集 R;用户集 U;操作集 Op;对象集 O;权限集 $P \subseteq Op \times O$;操作支配 $OpD \subseteq Op \times Op$;对象层次 $OH \subseteq O \times O$;权限支配 $PD \subseteq P \times P$;角色特化 $RS \subseteq R \times R$.其中 TH、OpD、OH、PD 和 RS 均为相应集合上的偏序关系.

定义1 复合职责 $TR \subseteq T \times R$,即任务-角色指派关系.$tr=(t,r) \in TR$ 表示角色 r 在任务 t 中承担某种职责.将 t 记为 $t(tr)$,r 记为 $r(tr)$.复合职责包含任务因素,在一定条件下可执行.若其可执行,其任务分量必须可执行.

定义2 关联特化 $TRS \subseteq TR \times TR$,$((t',r'),(t,r)) \in TRS \Leftrightarrow t' \leqslant t \wedge r' \geqslant r$,易证 TRS 是 TR 上的偏序关系.$tr' \geqslant tr$ 表示 tr' 和 tr 之间的特殊一般关系,称 $tr'(tr)$ 为 $tr(tr')$ 的特化(泛化)复合职责.任何泛化复合职责都是为便于管理引入的,故不可执行.

定义3 授权 $PA \subseteq P \times TR$.$(p,tr) \in PA$ 表示承担复合职责 tr 将可以获得权限 p.

定义4 用户指派 $UA \subseteq U \times TR$.$(u,tr) \in UA$ 表示复合职责 tr 可以由用户 u 来承担.

定义5 资源分配 $RA \subseteq U \times TR \times I$.$(u,tr,i) \in RA$ 表示指定用户 u 来执行任务实例 i 上的复合职责 tr,其前提是 tr 可执行,且其任务分量等于 $t(i)$.

2.2 访问检查

定义6 会话集记为 S.会话是用户与系统之间的交互过程,由用户发起.每个会话 s 对应于唯一的工作流案例,记为 $c(s)$,以及唯一的用户,记为 $u(s)$.

会话 s 在时刻 t 可激活的权限集 $auth\_perms(s,t) \subseteq \{p \in P \mid \exists\, tr \in auth\_trs(s,t) \cdot (p,tr) \in PA\}$,其中 $auth\_trs(s:S,t:\Gamma) = \{tr \mid \exists\, tr' \geqslant tr \cdot tr' \in trs(s,t)\}$,表示会话 s 在时刻 t 直接或间接激活的所有复合职责,而 $trs(s:S,t:\Gamma) \subseteq \{tr \mid \exists\, tr' \geqslant tr\ \exists\, i \in c(s) \cdot (u(s),tr',i) \in RA \wedge s(i,t)=1\}$,表示会话 s 中由 $u(s)$ 直接选择激活,且在时刻 t 仍处于激活状态的复合职责集.

3 职责分离框架

本节将定义包含角色、任务、权限和复合职责四种SoD的多维约束框架.

3.1 静态职责分离(Static SoD,SSD)

约束1 复合职责静态分离 $SSDTR \subseteq 2^{TR}$,$\forall\, trs \in SSDTR$,$\bigcap_{tr \in trs} auth\_users(tr) = \varnothing$,表示一组复合职责不

能有公共的成员.例如第 5 节案例中需求(*a*)~(*e*).

其中 $auth_users(tr:TRS)=\{u\mid\exists tr'\geqslant tr\cdot(u,tr')\in UA\}$表示有可能激活复合职责 *tr* 的所有用户.

约束 2　任务静态分离 $SSDT\subseteq 2^T$, $\forall ts\in SSDT$, $\bigcap_{t\in ts} auth\_users(t)=\varnothing$,表示一组任务不能有公共的参与者.例如第 5 节需求(*f*)、(*g*).

其中 $auth\_users(t:T)=\{u\mid\exists t'\leqslant t\ \exists r(u,(t',r))\in UA\}=\{u\mid\exists t'\leqslant t\ \exists r\cdot u\in auth\_users(t',r)\}$,表示任务 *t* 分解出的各种复合职责的所有成员.

约束 3　角色静态分离 $SSDR\subseteq 2^R$, $\forall rs\in SSDR$, $\bigcap_{r\in rs} auth\_users(r)=\varnothing$,表一组角色性质冲突,在任何任务中均不能指派公共的成员.例如第 5 节需求(*h*)、(*i*).

其中 $auth\_users(r:R)=\{u\mid\exists r'\geqslant r\ \exists t\cdot(u,(t,r'))\in UA\}=\{u\mid\exists r'\geqslant r\ \exists t\cdot u\in auth\_users(t,r')\}$,表示角色 *r* 所参与的各种复合职责的所有成员.

约束 4　权限静态分离 $SSDP\subseteq 2^P$, $\forall ps\in SSDP$, $\bigcap_{p\in ps} auth\_users(p)=\varnothing$,表示一组权限不能分配给同一个用户.例如为了避免财务欺诈,支票填写和账簿写入权限不能分配给同一用户.

其中 $auth\_users(p:P)=\{u\mid\exists tr\cdot u\in auth\_users(tr)\wedge(p,tr)\in PA\}$,表示有可能激活权限 *p* 的所有用户.

3.2　动态职责分离(Dynamic SoD,DSD)

约束 5　复合职责动态分离 $DSDTR\subseteq 2^{TR}$, $\forall s\in S$, $\forall t\in\Gamma$, $\forall trs\in DSDTR$, $|trs\cap auth_trs(s,t)|<|trs|$表示会话中任何时刻,一组复合职责不能被同时激活.例如为了保证数据完整性,(自动对账,会计)和(借贷登记,会计)不能在会话中同时激活.

约束 6　任务动态分离 $DSDT\subseteq 2^T$, $\forall s\in S$, $\forall t\in\Gamma$, $\forall ts\in DSDT$, $|ts\cap auth\_tasks(s,t)|<|ts|$,表示在会话中任何时刻,一组任务不能被同时激活.例如开会时段必须暂停工作,项目例会和测试任务动态分离,那么某人作为参与者激活了项目例会任务,则不能作为测试员激活测试任务.

其中 $auth\_tasks(s:S,t:\Gamma)=\{t\mid\exists r\cdot(t,r)\in auth\_trs(s,t)\}$,表示会话 *s* 在时刻 *t* 激活的任务集.

约束 7　角色动态分离 $DSDR\subseteq 2^R$, $\forall s\in S$, $\forall t\in\Gamma$, $\forall rs\in DSDR$, $|rs\cap auth\_roles(s,t)|<|rs|$,表示会话中任何时刻,一组角色不能被同时激活.例如例会主持人和项目经理角色动态分离,那么某会话在项目例会中激活了主持人角色,则不能在制订计划等任务中激活项目经理角色,以免干扰会议进程.

其中 $auth\_roles(s:S,t:\Gamma)=\{r\mid\exists t\cdot(t,r)\in auth\_trs(s,t)\}$,表示会话 *s* 在时刻 *t* 激活的角色集.

约束 8　权限动态分离 $DSDP\subseteq 2^P$, $\forall s\in S$, $\forall t\in\Gamma$, $\forall ps\in DSDP$, $|ps\cap auth\_perms(s,t)|<|ps|$,表示在会话中任何时刻,一组权限不能被同时激活.例如为了保证数据完整性,代码写入和编译脚本执行权限不能在会话中同时激活.

为便于表示,定义前/后缀通配符 $H\in\{S,D\}$, $X\in\{TR,T,R,P\}$,则所有动态 SoD 可记为 *DSDX*,等等.

4　约束冗余分析与检测

约束是权限系统必须满足的逻辑条件,它们之间可能存在逻辑蕴涵关系.若约束 $c_1\Rightarrow c_2$,称 c_1 覆盖 c_2,记为 $c_1\rhd c_2$.被覆盖的约束是语义冗余的.为了自动化冗余检测,必须对约束覆盖关系进行系统的分析.

本文的约束分析以良构性为前提.称一个约束为良构的,如果其目标元素之间不存在偏序关系.非良构约束可能干扰正常的授权.例如 $tr'>tr$,且$\{tr',tr\}\in SSDTR$,那么由于 *tr'* 的用户必然是 *tr* 的成员,该约束将造成 *tr'* 不能拥有任何成员,从而相应的职责无人承担.

4.1　约束覆盖判定规则

约束 1~8 之间的覆盖判定规则为:

规则 1(分量覆盖)

$\forall trs=\{tr_1,\cdots,tr_n\}\in 2^{TR}$,

$\forall ts=\{t_1,\cdots,t_n\}\in 2^T$, $\forall rs=\{r_1,\cdots,r_n\}\in 2^R$,

(1) $\forall 1\leqslant i\leqslant n\cdot t(tr_i)=t_i\Rightarrow ts\in HSDT\rhd trs\in HSDTR$

(2) $\forall 1\leqslant i\leqslant n\cdot r(tr_i)=r_i\Rightarrow rs\in HSDR\rhd trs\in HSDTR$

证明　(2)对 $H=D$ 的情形,用反证法.假设存在 $trs=\{(t_i',r_i)\mid 1\leqslant i\leqslant n\}\in 2^{TR}$使得 $rs=\{r_1,\cdots,r_n\}\in DSDR$ 而 $trs\notin DSDTR\Rightarrow\exists s\ \exists t\cdot trs\subseteq auth\_trs(s,t)$

$\Rightarrow\exists s\ \exists t\ \forall(t_i',r_i)\in trs\cdot(t_i',r_i)\in auth\_trs(s,t)$

$\Rightarrow\exists s\ \exists t\ \forall r_i\in rs\ \exists t_i'\in T\cdot(t_i',r_i)\in auth\_trs(s,t)$

$\Rightarrow\exists s\ \exists t\cdot rs\subseteq auth\_roles(s,t)$

$\Rightarrow rs\notin DSDR$,与 $rs\in DSDR$ 矛盾,结论得证.　证毕

规则 2(泛化覆盖)

(1) $\forall x\in xs\in 2^{TR}\cup 2^R\cup 2^P, x\geqslant x'$,

$xs\in HSDX\rhd(xs\cup\{x'\}-\{x\})\in HSDX$.

(2) $\forall t\in ts\in 2^T, t\leqslant t'$,

$ts\in HSDT\rhd(ts\cup\{t'\}-\{t\})\in HSDT$.

规则 3(子集覆盖)

$\forall x\subseteq x'\in 2^{TR}\cup 2^T\cup 2^R, x\in HSDX\rhd x'\in HSDX$

规则 4(传递覆盖)

$\forall p_1,\cdots,p_n\in P$, $\forall tr_1,\cdots,tr_n\in TR$, $\forall 1\leqslant k\leqslant n\cdot(p_k,tr_k)\in PA\wedge\{p_1,\cdots,p_n\}\in SSDP\Rightarrow\{tr_1,\cdots,tr_n\}\in SSDTR$.

规则 5(静态覆盖)

$\forall x\in 2^{TR}\cup 2^T\cup 2^R\cup 2^P, x\in SSDX\rhd x\in DSDX$.

4.2　判定规则完备性

本节将逐步证明规则 1~5 的完备性.

性质 1(转化有向性) 没有从 DSD 到 SSD 的覆盖.

由于访问检查不影响授权过程,性质 1 显然成立.

性质 2(偏序有向性) 在 1 – 8 每种约束内部,只存在沿 TRS, TH^{-1}, RS, PD 等偏序自下而上的覆盖.

证明 对约束 1,须证明若$\{tr_1,\cdots,tr_m\}\in SSDTR\triangleright\{tr_1',\cdots,tr_n'\}\in SSDTR$,则$\forall\ tr_i',\neg\ \exists\ tr_j, tr_j>tr_i'$.用反证法,假设$\exists\ tr_i',\exists\ tr_j, tr_j>tr_i'$,任取这样的授权方案,增加新用户 u,对所有 tr_k',令$(u,tr_k')\in UA$,则新方案违反$\{tr_1',\cdots,tr_n'\}\in SSDTR$.然而 $u\notin auth\_users(tr_j)$,否则必$\exists\ tr_k'\geqslant tr_j>tr_i'$违反良构性.因此该方案并不违反$\{tr_1,\cdots,tr_m\}\in SSDTR$.这与覆盖关系矛盾,结论得证.

对约束 5,须证明若$\{tr_1,\cdots,tr_m\}\in DSDTR\triangleright\{tr_1',\cdots,tr_n'\}\in DSDTR$,则$\forall\ tr_i',\neg\ \exists\ tr_j, tr_j>tr_i'$.用反证法,假设$\exists\ tr_i',\exists\ tr_j, tr_j>tr_i'$,任取这样的授权方案,类似约束 1 的情况,增加新用户 u 使其违反$\{tr_1',\cdots,tr_n'\}\in SSDTR$,并通过 RA 在同一工作流案例的任务实例中将$\{tr_1',\cdots,tr_n'\}$中的每个元素或其特化复合职责都分配给 u.在时刻 t,令用户 u 在会话 s 中直接激活 $tr_1',\cdots,tr_n'$,从而违反$\{tr_1',\cdots,tr_n'\}\in DSDTR$.然而 $tr_j\notin auth\_trs(s,t)$,否则必$\exists\ tr_k'\geqslant tr_j>tr_i'$违反良构性.此时并不违反$\{tr_1,\cdots,tr_m\}\in DSDTR$.这与覆盖关系矛盾,结论得证.

关于其他约束的结论可类似表述和证明. 证毕

性质 3(传递有向性) 不存在从复合职责分离指向其它实体分离,或者从其它实体分离指向权限分离的覆盖关系.

证明 例如不存在任务分离 ~ 权限分离的覆盖,根据性质 1 可知不存在从 $SSDT$ 到 $SSDP$ 的覆盖关系,还须证明的是:不存在 $t_1,\cdots,t_m,p_1,\cdots,p_n$ 使得

(1)$\{t_1,\cdots,t_m\}\in HSDT\triangleright\{p_1,\cdots,p_n\}\in HSDP$

(2)$\{t_1,\cdots,t_m\}\in SSDT\triangleright\{p_1,\cdots,p_n\}\in DSDP$

只需证明(2)不成立即可(例如,若存在$\{t_1,\cdots,t_m\}\in SSDT\triangleright\{p_1,\cdots,p_n\}\in SSDP$,则根据规则 5 可推出(2)成立,矛盾).反设存在 $t_1,\cdots,t_m,p_1,\cdots,p_n$ 使(2)成立,任取不违反该假设的授权方案,增加新的任务 $t_1',\cdots,t_n'$,角色 $r_1',\cdots,r_n'$和用户 u,不妨设 $m<n$,对任何 $1\leqslant i\leqslant n$,令 $tr_i'=(t_i',r_i')\in TR$,$(u,tr_i')\in UA$ 且$(p_i,tr_i')\in PA$(取 $p_n=\cdots=p_{m+1}=p_m$).若用户 u 在会话中同时激活 $tr_1',\cdots,tr_n'$,进而激活 $p_1,\cdots,p_m$,则其违反$\{p_1,\cdots,p_m\}\in DSDP$.但是,由于用户 u 和 $t_1,\cdots,t_m$ 均无关,新方案并不违反$\{t_1,\cdots,t_m\}\in SSDT$.这与覆盖关系矛盾,结论得证. 证毕

定理 1(分类完备性)

(1)规则 1 ~ 4 对 SSD 内部的覆盖判定完备.

(2)规则 1 ~ 3 对 DSD 内部的覆盖判定完备.

证明

(1)规则 2、3 对于 1 – 4 每种约束之内的覆盖判定是完备的.先证根据规则 2(1)和 3 可判定约束 1 上的任何覆盖.用反证法,设$\{tr_1,\cdots,tr_m\}\in SSDTR\triangleright\{tr_1',\cdots,tr_n'\}\in SSDTR$ 不能由规则判定,则必$\exists\ tr_i,\neg\ \exists\ tr_j',tr_j'\geqslant tr_i$.增加新用户 u,对任何 tr_j',令$(u,tr_j')\in UA$,则其违反$\{tr_1',\cdots,tr_n'\}\in SSDTR$.然而 $u\notin auth\_users(tr_i)$,故并未违反$\{tr_1,\cdots,tr_m\}\in SSDTR$.这与覆盖关系矛盾,结论得证.类似可证规则 2(1)和 3 对约束 3/4,规则 2(2)和 3 对约束 2 的覆盖判定完备,从而整个结论成立.

现在可证规则 1、2、3 对约束 2 ~ 约束 1 的覆盖判定完备.反设$\{t_1,\cdots,t_m\}\in SSDT\triangleright\{tr_1',\cdots,tr_n'\}\in SSDTR$ 不能由规则判定,则必$\exists\ t_i,\neg\ \exists\ tr_j'=(t_j',r_j'),t_i\geqslant t_j'$.增加新用户 u,对任何 tr_j',令$(u,tr_j')\in UA$,则所得方案违反$\{tr_1',\cdots,tr_n'\}\in SSDTR$.但由于 $u\notin auth\_users(t_i)$,并不违反$\{t_1,\cdots,t_m\}\in SSDT$.这与覆盖关系矛盾,结论得证.类似可证明规则 1、2、3 对约束 3 ~ 约束 1 的覆盖判定完备;规则 2、3、4 对约束 4 ~ 约束 1 的覆盖判定完备.

综上,规则 1 ~ 4 可判定约束 1 ~ 4 上所有可能的覆盖.

(2)规则 2、3 对于约束 5 ~ 8 的每种约束之内的覆盖判定完备;规则 1、2、3 对约束 6/7 ~ 约束 5 的覆盖判定均是完备的.证明类似于(1),但需在反证法最后一步构造会话激活过程以导出矛盾.类似易证不存在约束 8 ~ 约束 5 的覆盖关系.

综上知命题成立. 证毕

定理 2(转化完备性) 规则 1 ~ 5 对 SSD ~ DSD 的覆盖判定完备.

证明 由于权限的使用通过授权和访问两阶段来控制,若用户在授权阶段直接或间接地分配了一组实体(复合职责、任务、角色或权限),则有可能在访问阶段同时激活它们.通过 SSD 阻止用户同时激活一组实体的唯一方式是避免给用户分配该组所有实体.因此,若某个 SSD 约束覆盖某个 DSD 约束,则其必然覆盖此 DSD 对应的 SSD 约束(例如 $trs\in HDSDX$ 对应于 $trs\in HSSDX$),由定理 1(1),规则 1 ~ 4 对后者的判定完备,再使用规则 5 即可判定前者.于是,任何 SSD ~ DSD 的覆盖均可由规则 1 ~ 5 判定. 证毕

由定理 1、2 和性质 1 立得:

定理 3(完备性) 规则 1 ~ 5 对判定约束 1 ~ 8 上的覆盖关系是完备的.

推论 1(基数有向性) 不存在从基数较大的约束到基数较小的约束的覆盖.

证明　由完备性定理,约束 1 ~ 8 上可能存在的覆盖等价于由规则 1 ~ 5 可判定的覆盖.由于没有一条规则从基数较大的约束导出基数较小者,因此这样的覆盖关系无法根据规则 1 ~ 5 判定,从而不可能存在.

证毕

4.3　动态冗余约束检测

由于动态约束在访问阶段检查,其冗余会降低访问检查的效率.以下给出集中检测冗余 DSD 的未优化算法.设 E 表示 T、R、TR 或 P,而 ER 为 E 上的偏序关系,则算法时间为:$O(|DSD| * |HSDX| * |HSDX_k|^2 * |ER|)$.

算法 1　冗余动态职责分离检测

输入:安全管理员定义的约束和授权方案.

输出:所有冗余动态职责分离的集合 *rDSD*.

0.　$rDSD \leftarrow \varnothing$;

1.　/* 检测每种 *DSDX* 内部的冗余 */

　　//设 k,j 的增序与 *DSDX* 元素的基数降序一致

for (k = 1 to $|DSDX|$){

　for (j = 1 to $|DSDX|$){

　　//将 *HSDX* 中的第 k 个约束记为 $HSDX_k$.

　　if ($|DSDX_j| > |DSDX_k| \vee j = k$) continue;

　　//以 $\geqslant$ 表示 TRS, TH^{-1}, RS 或 PD

　　if ($\forall x \in DSDX_j \exists x' \in DSDX_k (x' \geqslant x)$)

　　{$rDSD \leftarrow rDSD \cup DSDX_k$; break;}

　}

}

2.　/* 根据 *DSDT* 检测 *DSDTR* 中的冗余 */

　　//设 k 的增序与 *DSDTR* 元素的基数降序一致,

　　// j 的增序与 *DSDT* 元素的基数降序一致.

for (k = 1 to $|DSDTR|$){

　for (j = 1 to $|DSDT|$){

　　if ($|DSDT_j| > |DSDTR_k|$) continue;

　　if ($\forall t \in DSDT_j \exists (t', r') \in DSDTR_k (t' \leqslant t)$)

　　{$rDSD \leftarrow rDSD \cup DSDTR_k$; break;}

　}

}

3.　类似于 2 根据 *DSDR* 检测 *DSDTR* 中的冗余.

4.　对于 DSD 中不在 *rDSD* 内的每个约束 $DSDX_k$,执行步骤 5 ~ 9;

5.　$r \leftarrow false$;

6.　若 $X = TR$ 或 $X = T$ 或 $X = R$,执行以下代码:

/* 根据 *SSDX* 判断 $DSDX_k$ 是否冗余 */

for (j = 1 to $|SSDX|$){

　if ($|SSDX_j| > |SSDX_k| \vee SSDX_j = SSDX_k$) continue;

　/* 以 $\geqslant$ 表示 TRS, TH^{-1}, RS 或 PD */

　if ($\forall x \in SSDX_j \exists x' \in SSDX_k (x' \geqslant x)$)

　　{$rDSD \leftarrow rDSD \cup DSDX_k$; $r \leftarrow true$; break;}

　}

}

7.　若 $X = TR$ 且 $r = false$,类似于 2 根据 *SSDT* 判断 $DSDX_k$ 是否冗余并设置 r.

8.　若 $X = TR$ 且 $r = false$,类似于 3 根据 *SSDR* 判断 $DSDX_k$ 是否冗余并设置 r.

9.　若 $X = TR$ 且 $r = false$,执行以下代码:

　　/* 根据 *SSDP* 判断 $DSDX_k$ 是否冗余 */

　　//设 j 的增序与 *SSDP* 元素的基数降序一致

for (j = 1 to $|SSDP|$){

　if ($|SSDP_j| > |SSDX_k|$) continue;

　if ($\forall p \in SSDP_j \exists tr' \leqslant tr \in SSDX_k (p, tr') \in PA$)

　　{$rDSD \leftarrow rDSD \cup DSDX_k$; break;}

}

10.　return *rDSD*;

5　案例研究

对文献[2]的软件项目案例略加扩展,其工作流和任务层次分别如图 2 和图 3,角色关系如图 4 所示.

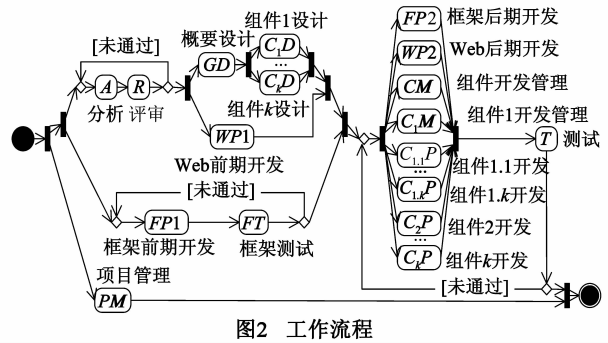

图2　工作流程

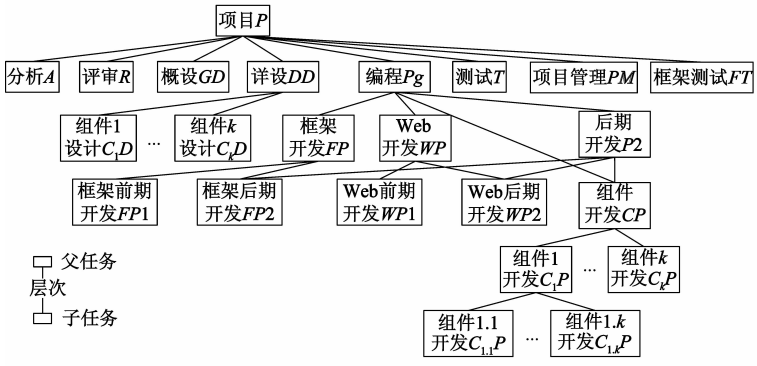

图3　任务层次

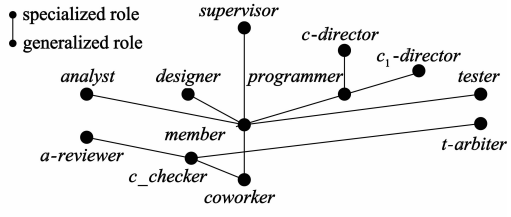

图4　角色特化关系

任务 *PM*、*A*、*R* 和 *FT* 分别由角色 *supervisor*、*analyst*、*a-reviewer* 和 *fp-tester* 负责,任务 *T* 由 *tester* 进行具体测试,由 *t-arbiter* 决定比较困难的 bug 能否挂起,*CM* 和 C_1M 分别由 *c-director* 和 c_1-*director* 负责,所有的设计和开发任务分别由 *designer* 和 *programmer* 负责,并且 *FP*1 还须由 *fp-reviewer* 进行代码审查.以上 *a-reviewer* 和 *t-arbiter* 来自客户方,其它角色属于项目内部.

案例的约束需求如下:

(*a*)内部成员承担工作,而客户方执行必要的检查,两者的人员互不相交.

(*b*)为保证需求分析的专业性及其评审的严格性,*A* 中的 *analyst* 和 *R* 中的 *a-reviewer* 不能由同一人担任.

(*c*)类似于(*b*),在 *Pg* 中担任 *programmer* 和在 *T* 中担任 *tester* 的人员不能有交集.

(*d*)类似于(*b*),在 *Pg* 中担任 *programmer* 和在 *T* 中担任 *t-arbiter* 的人员不相交.

(*e*)为避免测试人员轻率地挂起暂时难以修正的 *bug*,*T* 中的 *tester* 和 *t-arbiter* 不能由同一人员担任.

(*f*)为了保证评审对分析任务的检验作用,*A* 和 *R* 不能由同一人员执行.

(*g*)为了保证编程和测试工作各自的专业性,*Pg* 和 *T* 涉及的执行人员无交集.

(*h*)为了保证程序员和测试员岗位的专业性,担任 *programmer* 和 *tester* 的人员不能有交集.

(*i*)为保证测试工作的专业性以及挂起决策的权威性,*tester* 和 *t-arbiter* 不能由同一人员担任.

此外由于框架测试的特殊性,*fp-tester* 必须由开发人员担任,相应的约束需求是:

(*j*)在 *FP*1 中编程的人员不能承担 *FT* 任务.

(*k*) *FP*1 编程和代码审查不能由同样的人员进行.

相关工作以各种形式定义了任务或角色 SoD[1,2,9],但二者均无法描述团队任务相关的细粒度 SoD.对需求(*j*),若将 *FP*1 和 *FT* 定义为互斥任务,则框架前期开发的代码审查人员将不能执行框架测试任务,这是没有理由的;若将 *f-tester* 与 *programmer* 定义为互斥角色,则任何开发人员,特别是未参与框架前期开发的程序员都不能参与框架测试,导致该任务缺乏合适的人力资源无法执行.在本文中,令{(*FP*1, *programmer*),(*FT*, *f-tester*)},{(*FP*1, *programmer*),(*FP*1, *fp-reviewer*)} ∈ *SSDTR*,即可准确表达需求(*j*)和(*k*).

以三步授权机制为背景的基于逻辑程序约束语言有相当灵活的表达能力,但是它们均不能描述以上复合职责分离.文献[8]中采用二元谓词 *belong*(*u*, *r*)表示用户 *u* 是角色 *r* 的成员,role(*r*, *t*)表示角色 *r* 可以承担任务 *t*,用 cannot_do$_u$(*u*, *t*)表示用户 *u* 不能执行任务 *t*,等等.但是,它缺乏描述用户在任务中承担角色的三元谓词,也无法给出准确的替代.例如,尽管规则体中同时出现 belong(*u*, *programmer*)和 role(*programmer*, *FP*1)表明用户 *u* 有可能以程序员角色执行框架前期编程任务,但 *u* 并未明确获得该资格.若试图以规则 panic←belong(*u*, *programmer*),role(*programmer*, *FP*1),belong(*u*, *f-tester*),role(*f-tester*, *FT*)来描述(*j*),其实际约束效果将是:任何用户 *u* 都不能既是 *programmer* 又是 *f-tester*,或者 *programmer* 不能指派给 *FP*1,或者 *f-tester* 不能指派给 *FT*,均明显违反前述约束场景.若规则 cannot_do$_u$(*u*, *FT*)←belong(*u*, *programmer*),role(*programmer*, *FP*1)在某种程度上接近于(*j*)的涵义,那么以同样方式来描述(*k*)即可暴露其语义偏差:无论 cannot_do$_u$(*u*, *FP*1)←belong(*u*, *programmer*),role(*programmer*, *FP*1)或 cannot_do$_u$(*u*, *FP*1)←belong(*u*, *fp-reviewer*),role(*fp-reviewer*, *FP*1)都显然是不合理的.文献[7]的主要谓词与文献[8]类似,例如 can-play()对应于 belong(),而 hold()对应于 role(),出于同样原因也不能表达复合职责分离约束.此外,文献[5]也是描述能力很强的约束语言.在该文献中,任务互斥约束可表达为 $|\text{task}(OE(r,\text{role}(OE(u,U))))\cap OE(ct,CT)|\leqslant 1$(式中 *CT* 是互斥任务集合的集合),只有补充函数 task-role: $U\to 2^{TR}$ 才能表达复合职责分离的涵义.该文献还允许使用蕴涵符来定义规则,并且定义了 cannot_do$_u$()等谓词.但是,其规则也存在前述问题,例如对需求(*k*),将导致 *programmer* ∈ role(*OE*(*u*, *U*)) ∧ *FP*1 ∈ task(*programer*) ⇒ cannot_do$_u$(*OE*(*u*, *U*), *FP*1)这样语义错误的表达.总的来说,因为这些语言隐含着三步授权假设,缺乏描述复合职责与用户指派关系的谓词,必然无法表达(*j*)和(*k*)这样的需求.

(*a*)~(*i*)反映了软件项目中的多维度 SoD 需求,均可在本文约束框架中得到恰当描述,如表 1 所示.由于 *SSDTR* 的定义具有泛化特征,允许以粗粒度的约束涵盖若干条细粒度约束,因此 $trs_a\in SSDTR$ 即可简洁地表达需求(*a*).若 *SSDTR* 约束不可泛化,则必须指定{(C_1D, *designer*),(*R*, *a-reviewer*)},…,{(C_kD, *designer*),(*R*, *a-reviewr*)},…,{(C_1P, *programmer*),(*T*, *t-arbiter*)},… ∈ *SSDTR* 共计 $6k+18$ 条约束,表达复杂性大大增加.

根据前面的覆盖判定规则可以判定任何约束集中的所有冗余.例如表 1 约束包含的语义冗余有:

$ts_g\in SSDT\triangleright trs_c, trs_d\in SSDTR$ (规则 1(1))

$rs_i\in SSDR\triangleright trs_e\in SSDTR$ (规则 1(2))

$trs_a\in SSDTR\triangleright trs_b, trs_d, trs_e\in SSDTR$ (规则 2)

去掉 4 条冗余后,剩余 5 条约束彼此不存在覆盖关系,说明全部冗余可按规则判定.

表 1　职责分离约束

需求	约束	约束中的元素
(*a*)	*SSDTR*	trs_a:{(*P*, *member*), (*P*, *c_checker*)}
(*b*)		trs_b:{(*A*, *analyst*), (*R*, *a-reviewr*)},
(*c*)		trs_c:{(*Pg*, *programmer*), (*T*, *tester*)},
(*d*)		trs_d:{(*Pg*, *programmer*), (*T*, *t-arbiter*)}
(*e*)		trs_e:{(*T*, *tester*), (*T*, *t-arbiter*)}
(*f*)	*SSDT*	ts_f:{*A*, *R*}
(*g*)		ts_g:{*Pg*, *T*}
(*h*)	*SSDR*	rs_h:{*programmer*, *tester*}
(*i*)		rs_i:{*tester*, *t-arbiter*}

6　结束语

本文根据任务-角色指派及其关联特化关系定义了多维可泛化的职责分离框架,能够对团队任务相关的各种职责划分形式施加全面深入的约束,特别是可以描述细粒度的复合职责分离.随后系统分析了约束框架中的语义覆盖规则,为冗余约束检测、授权合理性验证等应用提供了理论依据,并给出动态冗余约束的检测算法.

下一步将对本文模型的委托机制进行研究.

参考文献

[1] S Oh, S Park. Task-role based access control model[J]. Journal of Information Systems, 2003, 28(6): 533 – 562.

[2] 翟治年,卢亚辉,等.增强伸缩性的主被动集成访问控制模型[J].计算机集成制造系统,2011,17(8):1609 – 1623.
Zhai Z, Lu Y, et al. Scalability enhanced active-passive-integrated access control model[J]. Computer Integrated Manufacturing Systems, 2011, 17(8): 1609 – 1623. (in Chinese)

[3] Wu S, A Sheth, J Miller J, et al. Authorization and access control of application data in workflow systems[J]. Journal of Intelligent Information Systems, 2002, 18(1): 71 – 94.

[4] 翟治年,奚建清,等.主动授权管理中的关联继承机制[J].西安交通大学学报(自然科学版),2012,46(4):24 – 31.
Zhai Z, Xi J, et al. Association inheritance mechanism in active authorization management[J]. Journal of Xi'an Jiaotong University (Natural Science), 2012, 46(4): 24 – 31. (in Chinese)

[5] 邢光林,洪帆.基于角色和任务的工作流授权模型及约束描述[J].计算机研究与发展,2005,(11):1946 – 1953.
Xing G, Hong F. A workflow authorization model based on role and task and constraints specification[J]. Computer Research and Development, 2005, 42(11): 1946 – 1953. (in Chinese)

[6] Wang Q, Li n. Satisfiability and resiliency in workflow authorization systems[J]. ACM Transactions on Information and System Security, 2010, 13(4): 1 – 35.

[7] J Wainer, P Barthelmess, A Kumar. W-rbac—a workflow security model incorporating controlled overriding of constraints[J]. International Journal of Cooperative Information Systems, 2003, 12(4): 455 – 485.

[8] E Bertino, E Ferrari, V Atluri. The specification and enforcement of authorization constraints in workflow management systems[J]. ACM Transaction on Information System Security, 1999, 2(1): 65 – 104.

[9] Liu D, Wu M, Lee S. Role-based authorizations for workflow systems in support of task-based separation of duty[J]. Journal of Systems and Software, 2004, 73(3): 375 – 387.

[10] 李凤华,苏▮,史国振,等.访问控制模型研究进展及发展趋势[J].电子学报,2012,40(4):805 – 813.
Li F, Su M, Shi G, et al. Research status and development trends of access control model[J]. Acta Electronica Sinica, 2012, 40(4): 805 – 813. (in Chinese)

[11] 王小明,付红,张立臣.基于属性的访问控制研究进展[J].电子学报,2010,38(7):1660 – 1667.
Wang X, Fu H, Zhang L. Research progress on attribute-based access control[J]. Acta Electronica Sinica, 2010, 38(7): 1660 – 1667. (in Chinese)

[12] 王雅哲,冯登国.一种 XACML 规则冲突及冗余分析方法[J].计算机学报,2009,32(3):516 – 530.
Wang Y, Feng D. Conflict and redundancy analysis method for xacml rules[J]. Chinese Journal of Computers, 2009, 32(3): 516 – 530. (in Chinese)

[13] V Kolovski, J Hendler, B Parsia. Analyzing web access control policies[A]. Proceedings of the 16th International Conference on World Wide Web[C]. Banff, Alberta, Canada, 2007. 677 – 686.

作者简介

翟治年　男,1977 年 5 月生,河北张家口人.2012 年毕业于华南理工大学计算机应用技术专业,获工学博士学位.现为浙江科技学院信息与电子工程学院讲师,主要从事信息安全与访问控制方面的研究.
E-mail: zhaizhinian@gmail.com

卢亚辉　男,1976 年 5 月生,陕西宝鸡人.1997 年、2003 年和 2008 年分别在南京大学、中科院遥感所和清华大学获理学学士、理学硕士和工学博士学位.现为深圳大学计算机与软件学院副教授,主要从事工作流、系统安全、Petri 网、Pi 演算等方面的研究.

奚建清　男,1962 年 7 月生,江苏江阴人,博士.现为华南理工大学计算机学院教授、博导,主要从事数据库与分布式系统方面的研究.