文章编号: 1000-6788(2012)12-2781-08 中图分类号: TP311 文献标志码: A

支持动态协作的委托授权意愿协商

王 媛[1], 孙宇清[1], 高荣瑞[2]

(1. 山东大学 计算机科学与技术学院, 济南 250101; 2. 中兴通讯股份有限公司, 深圳 518000)

摘 要 委托授权是为了实现资源共享与协作, 在主体之间进行的一种灵活授权方式, 是将自己拥有的某些权限转授给他人. 现有工作主要针对委托授权的时效性、单调性、委托深度、广度和粒度以及委托权限的撤销等方面进行研究, 提出了不同的委托授权模型和方法, 但是缺乏讨论委托授权中的意愿协商, 即同时支持委托者和受托者的意愿表达和匹配问题. 针对这一不足, 提出委托授权意愿逻辑, 描述委托授权双方意愿, 并通过模糊意愿匹配算法, 实现委托授权双方的意愿匹配; 提出委托授权协商模型, 将委托授权意愿逻辑与访问控制系统结合, 实验验证意愿匹配算法的有效性和效率.

关键词 访问控制; 委托授权; 协商

Supporting dynamic intention agreement for authorization delegation in multi-party collaboration

WANG Yuan[1], SUN Yu-qing[1], GAO Rong-rui[2]

(1. College of Computer Science and Technology, Shandong University, Jinan 250101, China; 2. Zhongxing Telecommunication Equipment Corporation, Shenzhen 518000, China)

Abstract Delegation is an assignment of authority and responsibility between persons for sharing resources and working collaboratively. Many models and solutions have been proposed with respect to delegation permanence, monotonicity, levels, granularity, and revocation etc. However, the important aspect of delegation agreement is less discussed which aims at carrying out delegation with consideration of the intention of both delegator and delegatee rather than the currently considering delegator only. In this paper, we propose delegation intention logic that is expressive, fine-grained and inferential in specifying the intentions of both sides and system security constraints. With the help of fuzzy logic, the proposed intention algorithm finds the most appropriate candidates for a delegation intention. Some experiments are performed to verify our method, as well as the implementation architecture is provided.

Keywords access control; delegation; agreement

1 引言

委托授权是指将自己拥有的某些权限转授给他人行使职责, 以实现主体间动态协作和资源共享. 委托授权是一种灵活的主体之间的授权方式, 适用于动态协作环境中解决临时性的、应急处理等方面的授权问题. 例如: 当某人临时不在岗位时, 他所负责的工作需要由他人代为实施, 这时需要将工作相应的权限暂时委托给他人. Barka 和 Sandhu[1−2] 提出了委托授权的基本特征被公认为实施委托授权时需要考虑的重要方面, 即委托持久性、单调性、完整性、委托粒度、委托深度、授权管理、委托协商、委托撤销等. 基于上述特征提出大量的委托授权模型, 主要讨论基于角色的访问控制模型 (RBAC) 中委托深度、支持细粒度授权等问题, 其中典型的委托授权模型有: 用户到用户委托授权模型 RBDM0[2]、支持角色层次和多步委托的委托授权模型 RDM2000[3]、基于许可的权限委托代理模型 PBDM[4] 等, 在 Wainer 给出的基于 RBAC 用户到用户委托授

收稿日期: 2011-07-25
资助项目: 国家自然科学基金 (61173140); 山东省自然科学基金 (Y2008G28); 山东大学自主创新基金 (2010JC010); 中国科学院计算机系统结构重点实验室开放课题基金 (ICT-ARCH200904)
作者简介: 王媛 (1982–), 女, 硕士研究生, 研究方向: 访问控制与隐私保护; 通信作者: 孙宇清 (1967–), 女, 教授, 研究方向: 系统安全与隐私保护, E-mail: sun_yuqing@sdu.edu.cn; 高荣瑞 (1985–), 男, 研究方向: 信息系统安全与访问控制.

权模型中 [5], 不仅考虑较高层次上的安全约束, 而且允许委托者制定委托条件. Crampton 研究授权管理并提出授权的授予与转移 [6]. 结合上下文环境, Wang 等提出访问控制系统中关于授权安全形式化概念 [7], 设计基于资源的实施机制来保证系统安全和效率. Atluri 等讨论工作流系统中的委托授权 [8], 引入限制概念作为委托需求, 例如: 时间、工作量、任务特点等.

但是, 上述工作主要针对委托者的委托需求, 没有考虑受托者的意愿, 受托者只能被动地接受, 属于单边委托, 这种方法适用于有严格用户层次等级的系统, 不适用于每个人都有权决定自己是否接受委托授权的情况. 例如: 某个教授想在自己开会期间将自己的权限授权给另一个教授, 这种授权不仅要考虑委托者的意愿, 还要考虑受托者的意愿, 即另一名教授是否愿意接受该授权. 因此, 有必要提供一种有效的机制, 支持委托授权双方意愿的表达, 并实现双方意愿的匹配.

针对现有的工作的不足, 提出委托授权意愿逻辑, 实现基于委托授权双方意愿的、多约束、细粒度、模糊的委托授权意愿的有效表达; 通过模糊意愿匹配算法, 从满足委托双方意愿的候选者中选择最适合的委托者或受托者, 实现双方意愿的匹配; 并将委托授权逻辑有机地整合到委托授权协商模型中, 实现动态协作的委托授权协商. 本文结构组织如下: 第 2 节描述委托授权意愿逻辑, 并对委托授权意愿逻辑进行性质分析, 第 3 节提出意愿匹配计算方法和模糊意愿匹配算法, 列举具体实例, 第 4 节简述委托授权协商模型及实验, 第 5 节总结本文工作和下一步研究分析.

2 委托授权意愿逻辑

有效的委托授权意愿逻辑表达, 需要满足以下性质: 1) 具有充分表达力, 在委托授权意愿表达中, 不仅支持静态的意愿约束要求, 同时还要支持系统状态的动态变化和环境描述. 2) 支持委托授权的细粒度表达. 3) 支持多种授权意愿并具有推理功能. 4) 委托授权意愿逻辑语言能够很容易、无误差的被转化为执行代码, 应用于实际系统. 5) 满足安全性, 支持委托者根据实际环境, 动态生成满足系统安全条件的临时委托角色集, 并通过细化约束集对授权制约, 避免违反系统安全的权限委托, 保证系统安全.

根据上述特点, 本文提出基于一阶逻辑的委托授权意愿逻辑, 用于委托授权双方意愿的表达, 并通过模糊意愿匹配算法, 将委托授权双方的意愿进行匹配. 在满足系统安全约束的范围内, 允许用户灵活地制定和修改意愿信息, 从而实现动态协作的委托授权意愿协商.

2.1 基本概念

委托授权意愿组成部分主要包括主体、客体、动作以及委托授权时需要满足的条件等. 委托授权的主体是指意愿的发布者即提出该意愿的用户, 所有主体的集合表示为 S; 客体是指意愿所涉及的委托授权内容, 如文件、数据等信息, 所有客体的集合表示为 O; 动作是指主体在客体上执行的操作, 如查看、修改等, 所有动作的集合表示为 A; 权限是指在某个资源上操作动作, 表示为 $\langle o, a\rangle$, 其中 $o \in O, a \in A$, 所有权限的集合表示为 P.

根据委托授权双方的属性以及系统环境属性来设置委托授权意愿, 其中每个属性称为意愿属性. 在委托授权意愿逻辑中用约束谓词 (constraint predicate) 来描述意愿属性, 表示为 $Predicate(x_1, x_2, \cdots, x_k)$, 其中 $Predicate$ 是谓词名, x_i 可以是变量、常量或者是一阶谓词, $i = 1, 2, \cdots, k$, 约束谓词的集合表示为 CP.

典型的约束谓词主要包括: 判定系统中是否存在的事实或者状态, 如判定主体 x 是否拥有教授角色, 表示为 $IsRole(x, \text{“}professor\text{”})$; 判定系统中是否执行某些动作或者改变某些状态, 如判定主体 x 是否参与过外科手术, 表示为 $Participate(x, \text{“}surgery\text{”})$.

定义 1 元约束 (constraint-element): 是指权限委托或执行委托时需要满足的基本条件和限制, 表示为: $Q_1x_1 \cdots Q_mx_m(e_1 \cdots e_n)$, 其中 $Q_i \in \{\exists, \forall\}$, x_i 表示实体变量, $i = 1, 2, \cdots, m$, 存在量词 “$\exists$” 表示系统中至少存在一个实体, 全称量词 “$\forall$” 表示系统中的所有实体; $e_j \in CP$, $j = 1, 2, \cdots, n$, 表示约束谓词.

根据约束内容的不同, 元约束可以分为三类:

1) 主体约束是指对委托授权主体的身份特征、角色、权限、资质等约束. 例如: 要求主体 x 是工作时间满 5 年的主治医生, 表示为: $\forall x\ IsRole(x, \text{“}AttendingDoctor\text{”}) \wedge IsLarger(WorkExp(x), \text{“}5Y\text{”})$;

2) 客体约束是指对委托授权的客体属性、安全等级、客体类型等约束. 例如: 要求客体 y 是心脏病人的病历, 表示为: $\forall y\ IsType(y, \text{“}Patient\_Record\text{”}) \wedge IsCategory(y, \text{“}Cardiopathy\text{”})$;

3) 环境约束是指对委托授权发生的系统状态、上下文环境等约束. 例如, 时间约束: 委托授权时间段为

[8:30AM, 11:30AM], 表示为: $TimeWithin(\text{“}8:30\text{AM”}, \text{“}11:30\text{AM”})$; 地点约束: 委托授权发生的位置约束, 表示为 $\forall x\ IsLocation(x, \text{“}office\text{”})$; 动作、事件约束: 考虑义务因素的授权约束, 如要求主体 x 在上午 8:30 之前检查监护病房, 表示为: $\forall x\ Examine(x, \text{“}ICUs\text{”}) \wedge TimeBefore(\text{“}8:30\text{AM”})$; 任务约束: 考虑受托者所能承受的最大工作量、任务强度、任务优先级、范围、任务执行时间等约束; 系统约束是系统在执行委托授权时必须满足的客观约束, 即任何委托授权的执行都不得违反高层次的安全约束, 如许可约束、职责分离、最小特权原则等.

2.2 委托授权规则

委托规则是指权限委托者设置的约束条件, 如受托者角色、委托时间、地点等限制, 选择满足自身意愿要求的受托者; 受托规则是指权限受托者提出受托意愿, 如对委托者身份、委托任务范围、执行时间等限制, 选择满足自身意愿要求的委托者.

定义 2 委托规则: $Can\_delegate[s, o, a] \leftarrow ce_1, ce_2, \cdots, ce_n$, 其中 $s \in S, o \in O, a \in A$, ce_i 为元约束, $i = 1, 2, \cdots, n$, 表示当且仅当受托者满足所有的元约束 ce_i 时, 委托者 s 愿意将指定的权限 $\langle o, a\rangle$ 委托给受托者.

定义 3 受托规则: $Accept\_delegate[s, o, a] \leftarrow ce_1, ce_2, \cdots, ce_n$, 其中 $s \in S, o \in O, a \in A$, ce_i 为元约束, $i = 1, 2, \cdots, n$, 表示当且仅当委托者满足所有的元约束 ce_i 时, 受托者 s 愿意接受委托者的委托权限 $\langle o, a\rangle$.

实例 1: Davis 是某医院的一名主治医生, 拥有角色 $AttendingDoctor$, 其权限集 $\{\langle Patient\_Record, write\rangle, \langle Patient\_Record, update\rangle, \langle Patient\_Record, read\rangle\}$. Davis 发布委托意愿: 在出差期间, 将自己的权限授权给有 3 个月以上工作经验的当班实习医生, 表示为: $Can\_delegate[Davis, Patient\_Record, \{read, write, update\}] \leftarrow \forall x\ IsRole(x, \text{“}InternshipDoctor\text{”}) \wedge IsLarger(WorkExp(x), \text{“}3M\text{”}) \wedge OnDuty(x)$.

Mary 是一名实习医生, 拥有角色 $InternshipDoctor$, 其权限集 $\{\langle Patient\_Record, read\rangle\}$, Mary 发布受托意愿: 愿意在自己当班的时候接受角色至少为主治医生的授权请求, 表示为: $Accept\_delegate[Mary, Patient\_Record, a] \leftarrow \forall x\ IsRole(x, \text{“}AttendingDoctor\text{”}), TimeWithin(\text{“}8:30\text{AM”}, \text{“}11:30\text{PM”}), a \in \{read, write, update\}$.

系统允许主治医生将其权限委托给他人, 满足系统安全条件, 根据委托授权双方意愿表达, Davis 和 Mary 的意愿相互匹配, 则在 Davis 离开后, Mary 就可以在规定时间内执行 Davis 的委托授权.

在上述的委托授权过程中, 委托授权双方都有着明确的委托授权要求, 必须严格满足委托授权约束才能实现委托授权, 这种约束称为严格约束. 但在实际应用中, 有些委托授权意愿无法明确的表达, 如要求受托者 “经验丰富”、“专业近似” 等; 或者委托授权双方对于委托授权约束允许在一定差异度范围内条件满足委托授权要求, 这种约束称为模糊约束. 为此采用模糊逻辑, 用于模糊意愿的表达与匹配.

2.3 模糊委托授权意愿

所谓模糊逻辑是将传统集合论中的绝对隶属关系模糊化, 模拟人们的思维方式来表示和分析不确定的信息. 通过对模糊集合中的每个对象的隶属关系来表示该对象属于集合的程度. 模糊谓词是基于模糊逻辑的谓词形式.

定义 4 模糊谓词 (fuzzy predicate): 形如 $\langle e, thrd\rangle$, 其中 $e \in CP$ 表示约束谓词, 实数 $thrd \in [0, 1]$, 表示该意愿属性匹配度的门限值.

将委托双方意愿匹配度与门限值比较, 当意愿匹配度 $\geq thrd$ 时, 表明比较对象满足该意愿发起者的意愿属性要求, 模糊谓词为真. 当 $thrd = 1$ 时, 表示该意愿属性必须完全匹配, 适用于严格约束; 当 $thrd = 0$ 时, 表示对该意愿属性不提出要求, 即所有用户均可以作为委托候选人. 当存在多个潜在的受托者时, 如果委托授权双方的意愿匹配度 $\geq thrd$, 认为该用户满足委托者的意愿要求, 将其作为受托候选者; 如果匹配度 $< thrd$, 表明该用户与委托者的要求差距过大, 不能进行委托授权.

在委托授权中, 一般涉及多个意愿属性的约束要求, 采用模糊委托、受托规则来表达委托、受托意愿.

定义 5 模糊委托规则: $Can\_delegate[s, o, a] \leftarrow \langle e_1, thrd_1\rangle, \langle e_2, thrd_2\rangle, \cdots, \langle e_k, thrd_k\rangle$, 其中 $s \in S, o \in O, a \in A$, $\langle e_i, thrd_i\rangle$ 为模糊谓词, $i = 1, 2, \cdots, k$.

模糊委托规则表示: 对于某个特定的委托意愿, 将委托双方对 e_i 表达的每个意愿属性的匹配度与对应的门限值进行比较, 若所有的模糊谓词均为真时, 受托者才能成为该意愿发布者的受托候选人.

定义 6 模糊受托规则: $Accept\_delegate[s, o, a] \leftarrow \langle e_1, thrd_1\rangle, \langle e_2, thrd_2\rangle, \cdots, \langle e_k, thrd_k\rangle$, 其中 $s \in S, o \in$

$O, a \in A, e_i \in CP$ 表示约束谓词, $thrd_i \in [0,1]$ 之间的实数, 表示意愿属性匹配度的门限值, $i = 1, 2, \cdots, k$.

模糊受托规则表示: 受托者提出对某个授权的受托意愿, 将委托双方对 e_i 表达的每个意愿属性的匹配度与对应的门限值进行比较, 若所有的模糊谓词均为真时, 表示受托者愿意接受委托者的权限 $\langle o, a\rangle$ 委托, 委托者成为该受托意愿的委托候选人.

2.4 委托授权意愿逻辑的性质分析

下面从意愿表达需求和描述语言性质方面分析本文提出的委托授权意愿逻辑, 并举例详细说明:

1. 可表达性: 能够有效整合系统状态、实体属性等相关因素. 例如, 委托者 Marry 提出授权给其他担任数据结构课程的教授修改大学一年级学生成绩单的委托意愿, 意愿中不仅对委托者的角色、担任的课程进行约束, 还对客体的类型进行限制.

委托意愿表示为: $Can\_delegate[Marry, student\_record, update] \leftarrow \langle IsRole(x, \text{“}Professor\text{”}), 0.8\rangle$, $\langle IsCourse(x, \text{“}Data\_structure\text{”}), 1\rangle$,$\langle IsCategory(y, \text{“}freshmen\text{”}), 1\rangle$.

2. 细粒度授权: 能够根据不同需求描述角色、权限、人物等不同层次的授权需求. 例如, 委托授权给实习医生读病人病历的权限, 表示为 $\langle Internship\_doctor, patient\_record, read\rangle$, 委托授权给主治医生修改病人病历的权限, 表示为 $\langle Attending\_doctor, patient\_record, update\rangle$.

3. 推理能力: 能够对用户的意愿进行细化与推理, 可以获取用户的潜在意愿, 例如当委托者同意受托者对数据库进行修改操作时, 必然也同意受托者对数据库执行读、写操作.

4. 可行性: 能够转换为可执行代码, 应用于实际应用系统.

5. 安全性: 能够描述高层安全需求. 例如, 职责分离安全约束可以表示为: $\forall x \ \forall \ yIsRole(x, \text{“}Account\text{”}) \wedge IsRole(y, \text{“}Audit\text{”}) \wedge \neg EQA(x, y)$.

综上, 本文提出的授权意愿逻辑能够较好地满足委托授权双方的意愿需求, 具有可行性, 同时能够表达系统安全的约束.

3 委托授权意愿匹配

3.1 意愿属性匹配度的计算

对于意愿属性匹配度的计算, 采用委托授权双方意愿属性两两匹配的方式进行比较, 通过将意愿属性差异值标准化, 获得意愿属性的匹配度. 首先, 计算意愿属性差异值. 为了将不同属性值进行统一模式表达, 采用 Han 等提出的差异值计算方法 [9], 计算两个意愿属性差异值, 记作 $f_e(v, v_i)$, 其中 v 和 v_i 分别表示有待比较的与约束谓词 e 相关的意愿属性取值. 通常, 前者 v 表示委托意愿中的意愿属性取值, 后者 v_i 表示有待比较的候选受托主体的意愿属性取值. 根据意愿属性取值类型的不同, 主要介绍两种意愿属性差异值的计算方法: 值域为集合的属性差异值计算, 通过比较集合中属性值的差异个数, 获得意愿属性差异值, 如角色属性等; 取值为连续数值的线性属性差异值计算, 通过比较意愿属性取值之间的线性差值, 获得意愿属性差异值, 如时间、年龄、工龄等属性.

意愿属性差异值计算公式:

$$f_e(v, v_i) = \begin{cases} \infty, & \text{若 } v = null \text{ 或 } v_i = null \\ 0, & \text{否则若 } v = v_i \\ \frac{a^{|v - v_i|}}{MAX - |v - v_i|}, & \text{其他} \end{cases} \tag{1}$$

其中 $i = 1, 2, \cdots, k$, v, v_i 为实数, MAX 是远大于 $|v - v_i|$ 的实数, a 是一个大于 1 的实数.

实例 2: 委托者 A 提出对受托者角色属性要求, 表示为: $\langle IsRole(x, \text{“}Professor\text{”}), 0.5\rangle$, 权限集合为 P= $\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$. 存在多名委托候选人 B, C, D, 其角色分别为副教授 Asso.Prof, 助理教授 Assi.Prof, 讲师 Lecturer, 其权限集合分别为 $P1 = \{p_1, p_2, p_3, p_4, p_5, p_7, p_{12}\}$, $P2 = \{p_1, p_2, p_3, p_4, p_{10}, p_{11}\}$, $P3 = \{p_1, p_2, p_{12}\}$; 利用集合属性差异值计算角色属性差异度的大小, 其中 v, v_i 分别表示集合 P, $P_i (i = 1, 2, 3)$ 的两个取值, $|v - v_i|$ 表示在集合 P 中但不在集合 P_i 中元素的个数, 设 MAX=100, a=2, 根据公式 (1) 计算求得 B, C, D 角色属性差异值分别为 $f_e(v, v_1)$=0.083, $f_e(v, v_2) = 0.337$, $f_e(v, v_3) = 1.376$, 数值越大, 属性差异值越大.

其次进行属性差异值标准化, 由于不同意愿属性取值不同, 采用差异值计算方法会造成计算结果差异大, 且不具有可比性, 需要对其标准化. 本文采用 sigmoid 函数进行标准化处理, 计算权限委托候选者与该意愿

的意愿属性匹配度, 记作 $M_e(v,v_i)$.

意愿属性匹配度计算公式:

$$M_e(v,v_i)=\begin{cases}0, & \text{若 } f_e(v,v_i)=\infty\\ 1, & \text{若 } f_e(v,v_i)=0\\ 1-\dfrac{1}{1+\mathrm{e}^{(-k)\times(f_e(v,v_i)-\frac{m}{f_e(v,v_i)})}}, & \text{其他}\end{cases} \tag{2}$$

其中 $f_e(v,v_i)$ 为意愿属性差异值, k,m 为非负且小于 1 的常数. $M_e(v,v_i)$ 的值域为 [0,1] 之间的实数, 且 $M_e(v,v_i)$ 随着 $f_e(v,v_i)$ 的减少而单调增加.

实例 3: 委托者 Alice 发布基于时间约束意愿, 表示为: $\langle TimeWithin(\text{“8:00AM”},\text{“11:00AM”}),0.6\rangle$, 受托者 Bob 希望执行授权的时间 [7:00AM, 9:30AM], David 希望执行授权的时间 [8:30AM–11:30AM], 则 $|v-v_1|=1.5$, $|v-v_2|=0.5$, 其时间属性取值表示如图 1 所示. 设 $a=2, MAX=100, m=0.1, k=0.1$, 根据上述公式计算求得 Bob 的时间属性匹配度 $M_e(v,v_i)=0.585$, David 的时间属性匹配度 $M_e(v,v_i)=0.669$, 则满足条件的委托者为 David.

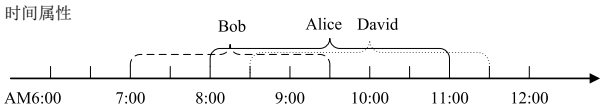


图 1 时间约束意愿属性

最后, 采用加权平均法计算整体意愿匹配度. 对于一条委托意愿表示为: $Can\_delegate[s,o,a]\leftarrow\langle e_1,thrd_1\rangle$, $\langle e_2,thrd_2\rangle,\cdots,\langle e_k,thrd_k\rangle$, 涉及多个元约束, 用权重表达每个元约束在整个委托意愿中的期望程度, 可以根据专家经验设置元约束权重矩阵, 记为 $W=(w_1,w_2,\cdots,w_k)$, 其中 $0\le w_i\le 1, i=1,2,\cdots,k, \sum_{i=1}^{k} w_i=1$, w_i 为权值系数, 表示某个意愿属性在整个委托元约束集中所占的比重, w_i 越高表示该元约束在整个委托意愿中越重要.

整体意愿属性匹配度计算公式:

$$r_i=(w_1,w_2,\cdots,w_k)\times\begin{pmatrix}M_{e_1}(v,v_i)\\ M_{e_2}(v,v_i)\\ \vdots\\ M_{e_k}(v,v_i)\end{pmatrix} \tag{3}$$

其中 $w_i\in W, i=1,2,\cdots,k$, $M_{e_i}(v,v_i)$ 表示权限委托候选者 s_i 对 e_i 表达的某个意愿属性匹配度, v 表示委托意愿属性取值, v_i 表示权限委托候选者 s_i 的意愿属性取值. 整体意愿匹配度 r_i 越大, 表示候选者 s_i 与主体意愿差异度越小, 越满足主体的要求, r_i 最大者作为最优候选者.

3.2 模糊意愿匹配算法

委托授权一般是由委托者作为行为的发起者, 本文提出了以委托者为中心的模糊意愿匹配算法, 从委托授权双方意愿中查找最合适的受托者. 具体步骤: 根据委托者意愿计算满足意愿的受托候选人, 对于不满足某个意愿属性匹配度要求的, 不继续进行意愿匹配度的计算, 按照候选人整体意愿属性匹配度的大小进行比较查找, 如果匹配值最大的受托者意愿中包含委托者, 则双方意愿匹配, 否则继续查找其他受托候选人, 直至双方意愿匹配为止.

算法: 模糊意愿匹配算法

输入: 一条委托意愿 d_i 和每个元约束的权值 $(w_1,w_2,\cdots,w_k)$, 受托意愿集合 $AI=\{a_1,a_2,\cdots,a_n\}$.

输出: 匹配结果集 $Result$.

1. $Candidate\_Queue=\emptyset$, $Result=\emptyset$//$Candidate\_Queue$ 为受托候选者队列
2. **for** $(j=1;j<=n;j++)$ **do**
3. **for** $(i=1;i<=k;i++)$ **do**
4. **if** $(w_i>0)$ **then**
5. **Calculate** $M_{e_i}(v,v_j)$ //计算受托者在委托意愿 d_i 中每个元约束 e_i 意愿属性匹配值 $M_{e_i}(v,v_j)$
6. **if** $(M_{e_i}(v,v_j)<thrd_i)$ **then**
7. **break**;

endif
endif
endfor
Calculate r_j //计算受托候选者的整体意愿匹配度 r_j
Insert_sort$(r_j, Candidate\_Queue)$ //按照 r_j 由大到小有序插入到候选受托者队列中
endfor
while (!$Candidate\_Queue.empty()$)
$Candidate\_Queue.Delete(a_i)$ //删除队头的受托意愿放至 a_i
for $(t = 1; t <= m; t++)$ **do** //m 为受托意愿 a_i 中元约束 e_i 的个数
Calculate $M_{e_i}(v, v_i)$ //计算委托者在受托意愿 a_i 中每个元约束 e_i 意愿属性匹配值 $M_{e_i}(v, v_i)$
if $(M_{e_i}(v, v_i) < thrd_i)$ **then**
break;
endfor
if $t = m + 1$ **then**
return $Result = \{a_i\}$//双方意愿匹配成功
endwhile
return $Result = \emptyset$//意愿匹配失败

3.3 实例分析

用系统状态表示用户、角色和权限授权关系的配置, 记为 $ST = \langle U, R, P, UR, PR\rangle$, 其中 U 是用户集, R 为角色集, P 为权限集, $UR = \{(u, r)|u \in U, r \in R\}$ 是用户角色授权关系, $(u, r) \in UR$ 表示 u 拥有角色 r, $PR = \{(r, p)|p \in P, r \in R\}$ 是角色权限授权关系, $(r, p) \in PR$, 表示角色 r 拥有权限 p, 某医院的系统状态如表 1 所示.

表 1 用户、角色、权限分配表

U	UR	PR
Alice	心脏病科室主任医生	$p_1, p_2, p_3, p_4, p_6, p_7, p_8$
Bob	骨科主任医生	$p_2, p_3, p_4, p_5, p_6, p_7, p_8$
Cathy	心脏病科室主治医生	$p_1, p_2, p_3, p_4, p_7, p_8$
David	骨科主治医生	$p_2, p_3, p_4, p_5, p_7, p_8$
Ellen	心脏病科室实习医生	p_2, p_7, p_8
Folw	骨科实习医生	p_2, p_7, p_8

注: R ={心脏病科主任医生, 骨科主任医生, 心脏病科主治医生, 骨科主治医生, 心脏病科实习医生, 骨科实习医生}; P ={p_1 心脏外科手术, p_2 阅读病例, p_3 收治病人, p_4 辅导实习医生, p_5 骨外科手术, p_6 组织医疗研究, p_7 使用医疗卫生系统, p_8 参与医疗工作}.

Alice 是该医院心脏病科室的主任医生, 希望在她休假期间委托授权给与自己角色、专业相近的当班医生, 表示为: $Can\_delegate[Alice, Patient\_Record, \{read, write, update\}] \leftarrow \langle IsRole(x, \text{“}AttendingDoctor\text{”}), 0.5\rangle, \langle IsProfessional(x, \text{“}Cardiopathy\text{”}), 0.4\rangle, \langle TimeWithin(\text{“}8:00\text{AM”}, \text{“}11:00\text{AM”}), 0.5\rangle$.

受托者 Bob 发布受托意愿: $Accept\_delegate[Bob, Patient\_Record, \{read, write, update\}] \leftarrow \langle IsRole(x, \text{“}ChiefDoctor\text{”}), 1\rangle, \langle TimeWithin(\text{“}8:30\text{AM”}, \text{“}11:00\text{AM”}), 0.6\rangle$.

受托者 Cathy 发布受托意愿: $Accept\_delegate[Cathy, Patient\_Record, \{read, write, update\}] \leftarrow \langle IsRole(x, \text{“}AttendingDoctor\text{”}), 0.5\rangle, \langle TimeWithin(\text{“}8:00\text{AM”}, \text{“}11:30\text{AM”}), 0.5\rangle$.

受托者 David 发布受托意愿: $Accept\_delegate[David, Patient\_Record, \{read, write, update\}] \leftarrow \langle IsRole(x, \text{“}AttendingDoctor\text{”}), 0.5\rangle, \langle TimeWithin(\text{“}8:00\text{AM”}, \text{“}11:30\text{AM”}), 0.5\rangle$.

设置意愿属性权值矩阵 $W = (0.4, 0.4, 0.2)$, 计算多个意愿属性的匹配度, 按照整体意愿匹配度大小进行排序, 其中 Cathy 的整体意愿匹配度最大, 同时受托者 Cathy 受托意愿集合中包含 Alice, 双方意愿匹配, 则在 Alice 离开后, Cathy 将在当班期间执行 Alice 的委托授权.

4　委托授权协商模型及系统实现

委托授权协商模型如图 2 所示, 它实现将委托授权逻辑有机地整合到委托授权模型和访问控制系统中. 模型主要包括三个方面 [10]: 1) 系统环境及安全访问控制; 2) 委托、受托意愿及一致性检测; 3) 委托受托双方意愿匹配. 委托者、受托者基于系统角色、权限等属性提出意愿请求, 其中意愿类型包含严格约束和模糊约束. 由于用户制定意愿的主观性和动态性, 可能会造成意愿间的相互冲突, 需要对意愿进行筛选, 将有效表达的双方意愿存放在意愿库中. 提取访问控制系统中各种安全约束信息和角色层次信息, 获取高层次的客观约束, 结合双方意愿进行意愿匹配, 若系统不存在满足意愿的候选者时, 将匹配失败信息反馈给委托者和受托者, 并适当地对委托或者受托意愿放宽限制, 从而实现委托授权协商.

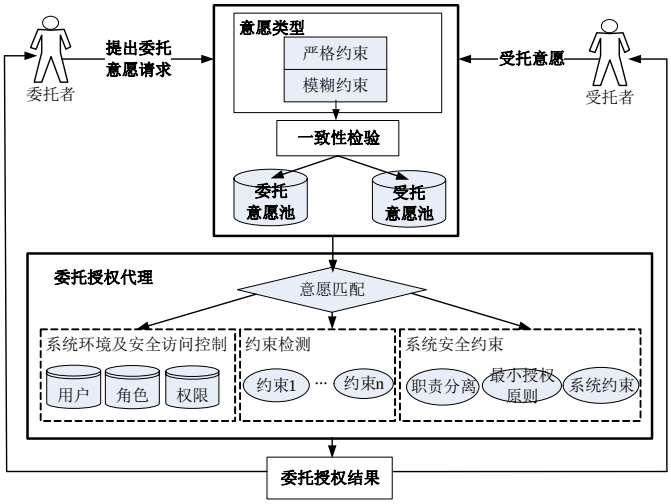
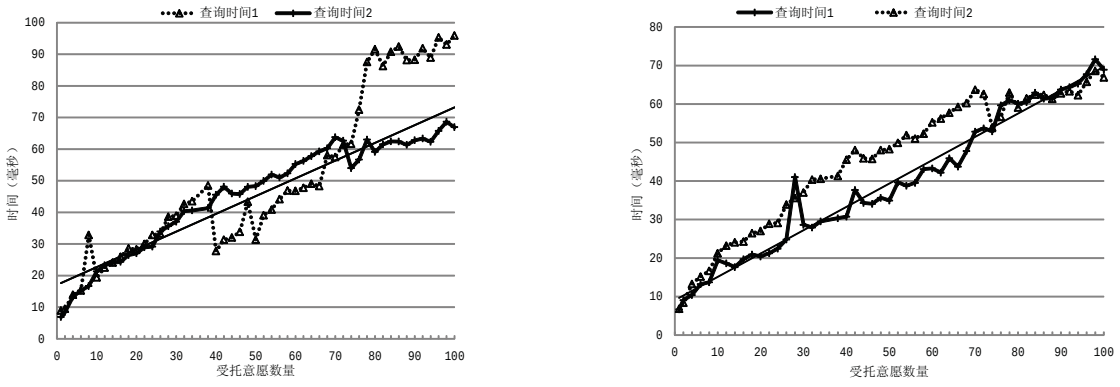


图 2　委托授权协商模型

为验证模糊意愿匹配算法的有效性, 根据委托授权协商模型设计意愿匹配系统. 实验环境为 CPU1.86GHZ, 内存 512M, 软件环境 Windows XP, 开发工具 MyEclipse+MySql. 建立意愿库, 设置意愿主体姓名、年龄、工龄、角色、专业等基本信息以及对每个意愿属性匹配门限值、权值等要求.

假设委托授权意愿库中受托意愿数量 1–100 个, 设置 5 个基本意愿属性, 其中年龄属性值区间为 25–65 岁, 工龄区间 1–40 年, 工作时间区间为 00:00–24:00, 专业分为骨科和心脏病科, 角色分为主任医生、副主任医生、主治医生、实习医生, 其数据分布为正态分布, 通过设置不同的意愿属性值、门限值和权值进行实验, 每次查询进行 100 次测试计算其查询时间的平均值, 共进行 50 轮, 得到实验结果如图 3 所示.



(a)门限值变化对意愿匹配时间的影响　　(b)权值变化对意愿匹配时间的影响

图 3　意愿匹配查询时间

图 3 反应了匹配查询时间与受托意愿数量的增长成正比, 基本上满足线性增长, 其中数据的波动是由于查询时间不仅受到受托意愿数量的影响, 还会受到数据分布的不均匀性以及门限值和权值的影响. 图 3(a) 中查询 1 是指所有意愿属性门限值均为零时的意愿匹配时间, 即不对意愿属性提出要求, 查询 2 是指设置多个

意愿属性门限值的意愿匹配查询时间, 该图表明意愿属性门限值对查询时间的影响, 增大委托意愿属性门限值, 其平均查询时间小于门限值均为零时的平均查询时间, 因为当门限值提高时, 满足委托者意愿属性的候选者人数会相应减少, 其查询时间也会相应减少; 图 3(b) 中查询 1 是指其中 2 个意愿属性权值为零时的意愿匹配查询时间, 查询 2 是指所有意愿属性均设定权值时的意愿匹配时间, 该图表明意愿权值对查询时间的影响, 将某些意愿属性数值设为零时, 即减少意愿属性的要求的个数, 其平均查询时间也随着意愿属性数量的减少而相应变小.

5 总结与展望

针对动态协作环境中委托授权缺乏双边意愿协商的问题, 提出了委托授权意愿逻辑用于描述委托双方意愿, 并引入模糊逻辑实现委托双方的模糊意愿表达, 增强委托授权意愿逻辑的语义表达能力. 提出意愿匹配算法, 综合考虑委托双方意愿, 设计委托授权协商模型, 将委托协商机制与访问控制系统有机整合; 进行了相关实验, 验证匹配算法的有效性. 下一步工作将优化委托授权协商的意愿匹配算法, 实现更为复杂环境中安全、有效的委托授权意愿协商.

参考文献

[1] Barka E, Sandhu R. Framework for role-based delegation models[C]// Proceedings of the 16th Annual Computer Security Application Conference, New Orleans: IEEE Computer Society Press, 2000: 168–176.

[2] Barka E, Sandhu R. A role-based delegation model and some extensions[C]// Proceeding of the 23rd National Information Systems Security Conference, Baltimore: IEEE Computer Society Press, 2000: 101–114.

[3] Zhang L H, Ahn G J, Chu B T. A rule-based framework for role-based delegation[C]// Proceeding of the 6th ACM Symposium on Access Control Models and Technologies, New York, USA: ACM Press, 2001: 153–162.

[4] Zhang X W, Oh S, Sandhu R S. PBDM: A flexible delegation model in RBAC[C]// Proceeding of the 8th ACM Symposium on Access Control Models and Technologies, New York, USA: ACM Press, 2003: 149–157.

[5] Wainer J, Kumar A. A fine-grained, controllable user-to-user delegation method in RBAC[C]// Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, 2005: 59–66.

[6] Crampton J, Khambhammettu H. Delegation in role-based access control[C]// Proceedings of 11th European Symposium on Research in Computer Security, 2006: 174–191.

[7] Wang Q, Li N, Chen H. On the security of delegation in access control systems[R]. CERIAS Technical Report, http://www.cs.purdue.edu/homes/wangq/papers/delegation.pdf, Jul. 2008.

[8] Atluri V, Warner J. Supporting conditional delegation in secure workflow management systems[C]// Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, 2005: 49–58.

[9] Han W L, Ni Q, Chen H. Apply measurable risk to strengthen security of a role-based delegation supporting workflow system[C]// 2009 IEEE International Symposium on Policies for Distributed Systems and Networks, 2009: 45–52.

[10] 高荣瑞, 孙宇清. 基于双边意愿的委托授权协商模型 [J]. 计算机科学, 2009, 36(9): 171–175.
Gao R R, Sun Y Q. A delegation agreement model based on bilateral intention[J]. Computer Science, 2009, 36(9): 171–175.