# 一种基于规则的委托约束授权模型

刘波 孙宇清

（山东大学计算机科学与技术学院，济南 250100）

**摘 要**：委托授权是访问控制模型的重要机制。现有的授权委托模型能够支持用户到用户、角色到角色的委托，并且支持层次角色环境下的委托，但是在这些模型中，在策略指定方面和对委托约束的研究还有待于进一步展开；并且在多人满足委托约束条件，如何对被委托人进行判定问题上存在不足。因此本文提出了基于规则的委托授权，引入了规则化的语言形式，从而通过判定约束规则，对委托进行判定。

**关键词**：委托；约束；访问控制

# A rule-based delegation constraint model

Liu bo Sun Yuqing

(School of Computer Science and Technology, Shandong University, Jinan, 250100)

**Abstract**: Delegation is an important mechanism of access control. The delegation models that have been present can support user-user and role-role delegation. And the models can support the delegation in role hierarchy. But in these models, about policy designation and delegation constraint need to farther study. When some users satisfy the delegation conditions, how to decide delegate is still a question. This paper provides a rule-based delegation model and inducts a regularization language form. Through determining the constraint rule, the paper introduces a evaluation mechanism to decide delegation.

**Keywords**: delegation; constraints; access control

## 1 引言

信息系统的广泛应用使得各企业组织对信息访问的安全性的要求越来越高。访问控制决定了谁能够访问系统的资源以及如何使用这些资源，适当的访问控制可以阻止未经允许的用户有意或无意的获取信息。访问控制的核心是授权策略，根据授权的方式不同可以分为直接授权和委托授权。直接授权是指通过指派，主体拥有访问某客体的权限；委托授权是指系统中的主体将其拥有的权限委托给其他主体，从而使该主体拥有对客体的访问权限。委托是访问控制模型的重要机制。它可以保证工作的继续执行；可以实现相互协作者之间资源和信息的共享[1]，因此具有重要的应用价值。

关于安全约束问题，目前存在的主要研究有 RBAC 委托中的不同约束[2]，从一个用户到另一个用户的委托只有在满足下面三种情况时才能发生：委托者本身有权利做出委托；委托满足所有委托者规定的限制；委托本身不能违反一般的约束。工作流中的委托[3]，约束分为授权约束，委托约束，任务依赖需求和角色激活约束；基于属性的委托模型[4]，该模型的委托条件是由先决条件（prerequisite condition）和委托属性表达式（delegation attribute expression）组成的，当给一个被委托者指派临时委托角色时，这个被委托者的先决角色和属性表达式必须同时满足委托条件的两个方面；受限制的委托模型[5]，该模型使用规则的表达式来约束委托树的形状，并且使用组信息来限制委托树，因此该模型对组信息需要一个中心的授权。但是这些模型中的约束都只是考虑在

某一个单一的应用场景下所遇到的限制，没有为用户提供一个较全面的约束参照，并且也没有考虑实际存在的另一个问题，即在满足委托约束条件下，如何对被委托人进行判定。

针对上述问题，本文提出了一个基于规则的委托约束授权模型，定义了多种委托约束，并将约束按照强制性和非强制性进行分类。对满足约束条件的被委托者进行判定委托。本文的结构如下，第二部分论述委托中的约束；第三部分将委托约束模型和其他的模型进行了比较；第四部分提出了委托约束模型的实验框架；第五部分结论和未来的研究工作。

## 2　委托模型

本文提出了委托约束授权模型。该模型是在 RDM2000 模型的基础上进行了改进，提出更加具体的约束分类，通过对委托约束的细化，使得委托执行更加安全有效，进一步避免了无授权实体对客体的访问。

### 2.1　基本模型及概念

如图 1 所示。

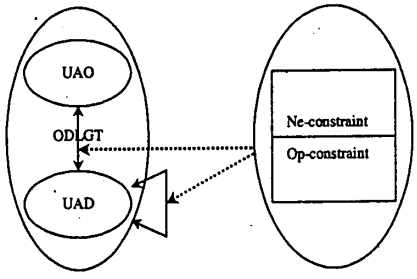


图 1　基本委托约束授权模型

图 1 中所用到的基本元素的定义如下：

P，R，U 分别代表权限集合，角色集合和用户集合

$UAO \subseteq U \times R$：初始用户到角色的指派关系

$UAD \subseteq U \times R$：委托用户到角色的指派关系

$UA \subseteq UAO \cup UAD$

$DLGT \subseteq UA \times UA = U \times R \times U \times R$：多对一的委托关系

$ODLGT \subseteq UAO \times UAD$：初始用户的委托关系

$DDLGT \subseteq UAD \times UAD$：委托用户的委托关系

$DLGT = ODLGT \cup DDLGT$

Ne-constraint：强制性约束，是各种强制性约束规则的组合

Op-constraint：非强制性约束，是各种非强制性约束规则的组合

Ne-constraint，Op-constraint 约束同时对 ODLGT 和 DDLGT 进行约束，我们将在下一节中具体介绍这两个约束。

其中初始用户 Original users 定义为通过用户—角色指派关系，直接指派给某一角色的用户；委托用户 Delegated users 定义为通过委托的用户—角色的指派关系，指派给某一角色的用户。

本文对委托模型中的约束进行了细化，将其具体分为强制性约束（ne-constraint），非强制性约束（op-constraint）。强制性约束是指委托时必须满足的约束，非强制性约束是指委托时可以选择的，并不一定是都包含的，根据具体情况而选的约束。关于强制性约束和非强制性约束的划分，是根据具体的情节而定的，一

般来说，约束力度比较高的可以作为强制性约束，而相对低的可以作为非强制性约束。

## 2.2 委托模型中的约束

委托是访问控制模型的重要机制，是指系统中的主体将其拥有的权限委托给其他的主体，从而使该主体拥有对客体的访问权限。委托不是任意发生的，必须对其进行有效的控制。对委托的控制可以从多个方面进行，例如对委托权限的控制，环境的影响，还有对被委托者本身的一些条件限制等。

许可约束：指用户是否有权对权限进行委托。当用户通过指派关系指派给角色，并且所要委托的权限也指派给该角色时，用户就有权对该权限进行委托。

我们用谓词 hasright(u,p)表示委托权利，是布尔变量。hasright(u,p)=1 表示用户 u 有权利对权限 p 进行委托，否则无权进行委托。

规则 1：$hasright(u,p) \leftarrow assign(u,r) \wedge assign(r,p)$

assign(u,r)代表用户和角色的指派关系；assign(r,p)表示角色和权限的指派关系。当两个指派关系都满足的时候，hasright(u,p)=1，否则为 0。

时间约束：表示只有在规定的时间内，才可以进行权限的委托，被委托者才可以执行委托的权限。规定了起始时间和终止时间。时间约束也可以用来对委托权限的撤销，只要超过 end-time，委托权限就自动撤销。

我们用谓词 timeconstraint（start-time，end-time）来表示时间约束。start-time 表示起始时间，end-time 表示终止时间。如果委托权限的时间在规定的范围内，那么 timeconstraint=1，否则为 0。

深度约束：表示是否允许被委托者将权限继续委托下去，具体包括以下几种深度的委托：①单步委托，既不允许委托，只有权限的拥有者可以进行委托，被委托者不能进行再委托。②多步委托，即允许被委托者进行再委托，但对在委托的步数进行限制，此时深度为一个正数，如 N,即许可被委托者再进行 N-1 步委托。③无限委托，即不对委托的深度进行限制，被委托者可以对权限进行再委托。

我们用谓词 depth（level,reg）表示，level 表示现在委托的深度，而 reg 表示规定允许委托的深度。例如 depth（1，4）表示现在的委托深度为 1，而允许的委托深度为 4，因此被委托者可以继续委托三步。随着委托的增加，level 自动加 1，而 reg 是固定不变的。

如图 2，假定 A→B，表明 A 将权限委托给 B；且在在角色层中，角色 A 的委托深度为 depth（0，2）。因此他可以继续委托 2 步，将权限委托给 B，这时 B 的委托深度为 depth（1,2），因此他可以继续向下委托 1 步，因此他把权限委托 C，此时 C 的委托深度为的 depth（2，2），不能在继续向下委托。

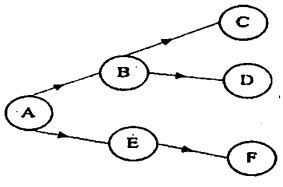


图 2 多步委托

规则 2：$depth(level,reg)=1 \leftarrow level \leq reg$

规则 2 说明：只有在 level≤reg 时，只有当委托深度小于委托限制的深度时，委托才为真。level≤reg 说明，进行的委托传递不能超过规定的最大委托深度。

图中表明 A 可以把权限委托给 B 或 E 或同时给 B 和 E，在这里我们不考虑委托广度，认为 A 不能把权限即委托给 B 又委托给 E。

职责分离约束：职责分离是委托安全的重要保证，被广泛的认为是计算机安全的基础原则。在 RBAC 中，职责分离是通过要求指派给被委托者的权限不能和他原来本身具有的权限发生冲突的方法实现的。我们用谓词 Csod 权限之间职责分离，是一个布尔变量，若满足职责分离的要求，Csod=1，否则为 0。

工作量约束：控制被委托者的工作量不能超过一定值。用谓词 canaccept（U，wl）来表示主体能承担委托的工作。规定 CW 为一阈值，表示 U 所能承担的最大工作量是被委托者自己规定的。

我们定义两个谓词 init_workload(u,wl)和 de_workload(u,wl)。其中 init_workload(u,wl)表示主体 U 初始的工作量为 wl；de_workload(u,wl)表示主体 U 委托的工作量为 wl。

根据上述的三个谓词定义，我们得出对工作量约束的规则。

规则 3：$canaccept(u,wl')=1 \leftarrow init\_workload(u,wl)+de\_workload(u',wl') \le cw$

规则 3 说明接受委托的主体，其原有的工作量加上委托的工作量不能超过其所能承担的最大工作量。

位置约束：控制委托只能发生在预定的位置，像委托者的办公室或者是和委托者的房间里的电脑等。用谓词 local（u）表示主体所在的位置，CL 表示规定的委托发生的位置。

规则 4：$localconstraint(u)=1 \leftarrow local(u)=cl$

规则 4 中的“=”代表语义上的相一致。

对上述介绍的六种约束，我们可以根据约束力度的大小进行分类。约束力度大的为强制性约束，表示委托时必须满足的约束（necessary_constraint）；对于约束力度小的为非强制性约束（optional_constraint），指委托时可以根据具体的应用背景选择的，并不一定是都包含的。

表 1 中的时间约束既可以为强制性约束也可以为非强制性约束，可以根据不同的应用背景由系统管理员进行设定。

表 1 约束分类

| 约束名称 | 约束力度 | 约束分类 |
|---|---|---|
| 许可约束 | 强 | 强制性 |
| 时间约束 | 中等 | 强制性/非强制性 |
| 深度约束 | 强 | 强制性 |
| 职责分离约束 | 强 | 强制性 |
| 工作量约束 | 弱 | 非强制性 |
| 位置约束 | 弱 | 非强制性 |

## 2.3 委托规则

我们用一种规则化的语言形式来表示委托约束，这样可以使委托约束的表达简单明了。

定义：规则形式

$H \leftarrow F1 \& F2 \& ... \& Fn$

其中 $H$，$F1$，$F2$，$...Fn$ 是一些布尔函数，&表示与的意思。规则表示根据 $F1$，$F2$，$...Fn$ 推理出 $H$。基本的委托授权规则形式为 $H \leftarrow$。

规则 5：强制性约束规则

$$ne\_constraint(u,p,u') \leftarrow hasright(u,p) \& depeth(level,reg) \& Csod$$

规则 6：非强制性规则

$$op\_constraint(u,p,u') \leftarrow F1 \& F2 \& ... \& Fn$$

由于非强制性约束是根据应用背景具体选择的约束，因此对于非强制约束没有特别规定的规则形式，但是规则中各个元素的关系是&，$F1$，$F2$，$...Fn$ 可以是位置约束，工作量约束和时间约束的任意组合。

规则 7：用户到用户的委托授权规则

$$can\_delegate(u,p,u') \leftarrow ne\_constraint(u,p,u') \& op\_constraint(u,p,u') \& maxt(u')$$

其中 u 代表委托者，p 代表委托权限，u′ 表示被委托者。

这个规则是基本的用户委托授权策略，表示用户 u 可以把权限 p 委托给用户 u′。

我们用一个医院购买电子医疗设备的实例对委托的过程进行简单说明。一个医院需要购买一批电子医疗设

备，该医院规定只有这个医院的采购部门的经理有购买设备的权利，并且如果这个经理出差了可以委托这个购买权限，但是只能委托给他的下一级，且只能委托一次，要求被委托人的工作量不超过 wl。

根据应用背景我们可以得出，强制性约束用到的是：许可约束，深度约束，职责分离约束；非强制性约束用到的是：工作量约束。如果副经理 1 和副经理 2 向采购经理提出申请，采购经理根据规则对其进行判定，从而选定被委托者。

## 3 比较

上述的委托约束模型是一种基于委托约束规则的授权模型，我们将从委托的程度，委托的控制，以及是否支持用户——用户委托和角色——角色委托，访问控制的安全性五个方面，对委托约束模型和目前存在的 RBDM0，RDM2000，PBDM 进行比较。委托约束模型对委托约束的要求更具体化，从而使委托的程度更加细粒度化；同时由于规则的定义和使用，使得对委托的控制更加得简单和明了，也更加有力；委托约束模型支持用户—用户的委托，模型不能实现角色——角色的委托，这也是将来的研究内容；由于委托约束模型的细粒度化和对委托的有力控制，因此委托约束模型能够更加有效的组织未授权用户对资源的访问，具有更高的安全性。如表 2（表中√√√表示模型在这一方面的程度最高，√√次之，√表示最小；空格表示模型不支持或者不具备这一方面能力。

表 2 委托约束模型与其他模型的比较

| | RBDM0 | RDM2000 | PBDM | 委托约束模型 |
|---|---|---|---|---|
| 委托的细粒度化 | | | √√ | √√√ |
| 委托的控制 | | √√ | √√ | √√√ |
| 用户—用户委托 | √ | √ | √ | √ |
| 角色—角色委托 | | | √ | |
| 安全性 | √ | √√ | √√ | √√√ |

## 4 实验框架

图 3 给出了委托约束模型的体系结构。主体首先发出委托请求，授权代理将主体的请求发送给访问控制代理，访问控制代理根据角色服务器中的数据库判断主体的角色，角色数据库向访问控制代理进行响应，如果是角色数据库中的角色，访问控制代理将主体的请求发送给委托代理，由委托代理进行最后的判定。参数监管的作用是对访问控制代理和委托代理进行能够审计和监管。委托代理通过询问规则服务器来判定委托是否有效。其中数据库中的规则就是我们上述中的各种委托约束规则，我们采用 XML 将约束规则在数据库中进行描述。最后，授权代理将委托代理的响应返回给主体。

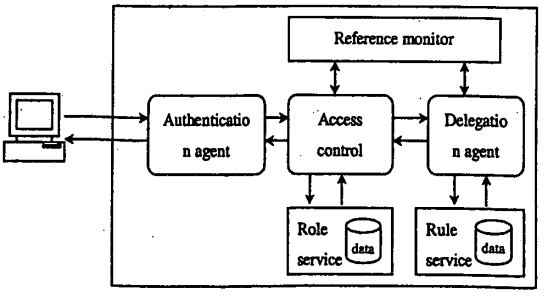


图 3 委托约束模型的体系结构

## 5 结论和未来的工作

本文主要研究了基于规则的委托授权，将授权委托中可能遇到的约束条件进行分类，并用谓词加以表示，从而形成一定的规则，并将文中提出的模型和已经出现的模型进行了比较，最后提出了委托约束模型的体系结构。

未来的研究工作：首先本文是研究授权者直接对被授权者的信任的判断进行权限委托，但是在大规模的异构的分布式系统中，系统的授权者无法直接知道用户，因此需要通过第三方获得信息，而授权者对第三方的信任是有一定程度的，这种基于第三方信任的分布式委托授权方式是未来的研究工作；其次本文只是从理论上对约束进行分类并没有进行实验验证，以后将通过实验验证这种分类规则的可行性；最后本文对于委托撤销采用的是确定超时撤销机制，没有考虑用户的撤销，这也将是研究的一个方向。

### 参考文献


[1] X. Zhang, S. Oh, and R. Sandhu. PBDM: a flexible delegation model in RBAC. In Proceedings of eighth ACM SACMAT, pages 149 - 157, 2003.

[2] J. Wainer and A. Kumar. A fine-grained,controllable,user-to-user delegation method in rbac. In Proceedings tenth ACM SACMAT, pages 59 - 66, 2005.

[3] V. Atluri and J. Warner. Supporting conditional delegation in secure workflow management systems. In *Proceedings Tenth ACM SACMAT*, pages 49 - 58, 2005.

[4] Chunxiao Ye and Zhongfu Wu. Using XML and XACML to Support Attribute Based Delegation. In *proceedings of the 2005 The Fifth International Conference on Computer and Information Technology*, 2005.

[5] Olav Bandmann, Mads Damy, Babak Sadighi Firozabadi, Constrained delegation. In *Pro-ceedings of the 2002 IEEE Symposium on security and privacy*, 2002.

[6] Gang Yin, Huai-min Wang, Dian-xi Shi, Yan jia, and Meng Teng. A rule-based framework for role-based constrained delegation. In *Infosecu'04*.

[7] Eric Freudenthal, Tracy Pesin and Lawrence Port. Drbac:Distributed role-based access control for dynamic coalition environments. In *Proceedings of the 22ndinternational conference on distributed computing systems*, 2002.

[8] Longhua Zhang, Gail-Joon Ahn and Bei-Tseng Chu. A rule-based framework for role-based delegation. In *SACMAT'01*, pages 153 - 162, 2001.

[9] Gang Yin, Huai-min Wang, Dian-xi Shi, Yan Jia and Meng Teng. A rule-based framework for role-based constrained delegation. In *InfoSecu'04*, pages 186 - 191, 2004.

[10] Nathan Griffiths. Task delegation using experience-based multi-dimensional turst. In *AAMAS'05*, pages 489 - 496, 2005.

[11] C. Castelfranchi and R. Falcone. Principles of trust for MAS:Cognitive anatomy, social importance, and quanti_cation. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, pages 72 - 79, 1998.

[12] D. Gambetta. Can we trust trust? In *D. Gambetta, editor, Trust: Making and Breaking Cooperative Relations*, pages 213 - 237, Basil Blackwell, 1988.

[13] S. Marsh. Formalising. Trust as a Computational Concept. *PhD thesis, University of Stirling*, 1994.

[14] E. Triantaphyllou. Multi-Criteria Decision Making Methods: A Comparative Study. *Kluwer Academic Publishers*, 2000.

[15] Konstantina Stoupa, Athena Vakali, Fang Li and Ioannis Tsoukalas. XML-based revocation and delegation in a distributed environment. In *EDBT 2004 Workshops*, pages 299 - 308, 2004.

# 一种基于规则的委托约束授权模型

作者： 刘波， 孙宇清
作者单位： 山东大学计算机科学与技术学院，济南，250100