单位代码	10445
学　　号	2011021089
分 类 号	TP393
研究生类别	全日制硕士

硕　士　学　位　论　文

论文题目：基于属性 RBAC 的访问控制模型研究

学科专业名称:　计算机软件与理论

申 请 人 姓 名:　李阳

指　导　教　师:　宋承祥

论文提交时间:　2014 年 4 月 16 日

独 创 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得_____（注：如没有其他需要特别声明的，本栏可空）或其他教育机构的学位或证书使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

学位论文版权使用授权书

本学位论文作者完全了解 **学校** 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 **学校** 可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。（保密的学位论文在解密后适用本授权书）

学位论文作者签名：　　导师签字：

签字日期：2014　年　　月　　日　　签字日期：2014　　年　月　日

目 录

摘 要

近几年，访问控制作为实现网络安全的一种技术措施渐渐成为研究热点。传统的访问控制模型及其扩展模型有很多，其中比较常见的有:自主访问控制模型（Discretionary access control model）、基于角色的访问控制模型（Role based access control model）、强制访问控制模型（Mandatory access control model）、基于属性的访问控制模型（Attribute based access control model）。在这之中使用控制模型（Usage control model）是较为完善的一种访问控制模型，不过它虽然改进了传统访问控制模型在授权和委托方面的缺陷，但在控制策略管理和细粒度划分方面依然有不足之处。较为详尽的剖析了访问控制的研究现状以及其核心技术分析了开放环境中解决安全威胁的迫切性、实现系统安全访问的必要性，陈述了访问控制模型的发展历程、目前的研究现状及核心技术，对比剖析了各种模型的优势及缺陷，阐述了当前开放环境的特点等。本文围绕 UCON 模型的优缺点，对控制模型进行了探索和研究，主要的内容以及创新点如下：

1.建立了一种基于属性 RBAC 的访问控制模型。

针对传统使用控制模型不能够控制属性、无法实现安全的委托授权而导致的访问控制不灵活问题，提出一种新的基于属性 RBAC 的访问控制模型。在理论层面分析各组成要素的逻辑关系，并提出了与其配套的使用控制策略模型。在此模型中，角色、属性及使用控制决策因素义务、条件等相结合，把属性和角色的授权委托的模块嵌入 UCON，实现了委托的功能，提高了灵活性和可靠性。

2.建立了一种具有时间约束的跨域控制模型

当下网络的发展趋势为开放、异构的，系统间的跨域访问操作越发频繁，跨域的访问控制研究很有必要。另一方面，在网络系统中，依据时间段来控制访问亦越发的普遍。按照目前的网络系统现状，将时间约束和源域及目标域概念引入使用控制模型，提高了模型的适应性和实用性。

3.设计仿真实验验证模型的实用性

通过智能卡访问管理实例，验证本文提出的控制模型的实用性。在线阅读实例中，通过一个用户将部分在线书籍的阅读权转授予另一个用户，验证本文提出的控制模型对授权委托的控制。

关键词：访问控制，RBAC，UCON,使用控制决策

ABSTRACT

In recent years,Network application technology development,Network with people's life more and more to close.The security problem of network is getting more and more attention.control as a technical measure to realize the network security has gradually become the research hot spot,The traditional access control model and its extended model has a lot.Among the more common are Discretionary access control model（DAC Model）,Role based access control model（RBAC Model）,Mandatory access control model (MAC Model),Attribute based access control model（ABAC Model,Usage control model (UCON Model) is a kind of access control model more perfect,But although it has improved the traditional access control model and the principal aspects of the defect,But there are still shortcomings in the aspects of control strategies of management and fine granularity partition,In this paper,the advantages and disadvantages of the UCON model,The exploration and research of control model,The main contents and innovations are as follows

1.a detailed analysis of the research status of access control as well as its core technology

This paper analyzes the urgency to solve the security threat in open environment,The necessity of implementation of security access.Statement of the development history of access control model,Research status and key technology at present,Comparative analysis of the advantages and disadvantages of various model,Describes the current open environment characteristics

2.to establish a RBAC attribute based access control model

In view of the traditional usage control model can not control attribute,Unable to achieve security authorization and access control problems lead to inflexible,Presents a new access control model based on attribute RBAC.Analysis of the logical relationship of each component in the theoretical level And proposes the use of a matched control strategy model,In this model,Characters,properties and use control decision factors,combined with duty conditions,The attribute and role authorization module embedded in UCON,The principal functions, improving the flexibility and reliability

3.to establish a temporal cross domain control model

The current development trend to open,heterogeneous network,The system of cross domain access operation is more and more frequently,It is necessary to study the cross domain access control.On the other hand，In the network system,On the basis of period of time to

control access is also increasingly common.According to the situation of the current network system,The time constraint and the source domain and the target domain concept of usage control model,To improve the adaptability and practicability of the model

4.The practical design and Validate the practicability of model

Practical control model verification is presented in this paper Through the examples of smart card access management,Reading on-line example,Model verification is presented in this paper to control the authorization Through a user will be part of the online book reading delegated to another user

Key Words： Access control,RBAC,UCON,Use the control decision

第一章 绪 论

1.1 研究背景及意义

全球互联网技术的飞速发展以及全球信息交互的频繁，人们的生活交流变得更加的快捷高效。但在享受便捷生活之余，也不得不面临企业、个人隐私机密数据信息容易泄露、丢失甚至被恶意破坏的安全威胁。访问控制作为解决这一问题的一种技术手段成为了一个重要的研究领域[1]。而随着网络交互，云计算、物联网的技术的不断发展，分布式系统逐渐取代单一集中式的计算机系统，系统交互协作与越来越普遍的网络应用服务等不断考验着访问控制及其授权管理。访问控制服务是五大安全服务功能之一，它具有预防非授权用户非法进入系统、限制客体资源访问、引导其它安全服务的作用，从而保护整个系统不被非法进入及非法操作[2]。中国也将访问控制定位成保护系统安全的重要措施之一，几乎包含所有安全等级，其重要性可见一斑[3]。

基于差异性的信息系统，访问控制模型也有很多。比较多见的访问控制模型主要有以下几种：分别是自主访问控制模型（DAC）[4]、强制访问控制模型（MAC）[5]、基于角色的访问控制模型（RBAC）[6]、基于属性的访问控制模型（ABAC）[7]和使用控制模型（UCON）[8]。19 世纪 70 年代初，访问控制技术被提出。最开始是为了实现大型计算机系统共享数据授权访问的安全性。访问矩阵作为早期访问控制技术的雏形被用于操作系统中。随着矩阵的标准化，便形成了早期的访问控制模型，也就是自主控制模型（DAC）[9]。此模型中，主体应用者兼任其创建者，能主动将自己的访问操作权授给其他主体，使其获取对系统资源的访问控制权。与此同时，强制访问控制模型也被提出，此模型的核心思想是系统中的资源（包括系统中的数据、用户）按相应的安全等级进行分类[10]。控制信息流动的方向只能是从低级到高级，从而避免因高级的数据流向低级区域而造成数据泄漏，进而实现对系统数据的安全访问控制。

网络技术的不断创新与发展，越来越多的应用与数据交换通过网络来实现，传统的一些访问控制技术无法满足现行系统的安全要求，在这样的背景下，新的访问控制模型应运而生。RBAC 于 1996 年被形式化提出，此模型首次提出了角色的概念，并将其作为桥梁联系起用户与权限，是授权操作更加的灵活[11]。权限不再像传统访问控制那样被直接授予用户，而是通过先将对特定客体访问等操作的权限包含在某一个角色中，然后通

过授予用户角色来实现用户对系统中相应客体资源的访问等操作，这使得基于角色的访问控制模型拥有比自主和强制访问控制模型更加灵活的授权操作及系统适应性[12]。第二年，基于角色的访问控制模型的管理模型被提出，即 AdministrativeRBAC。它构成了基于角色的一个经典模型（也称 RABC96 模型）[13]，对以后的访问控制模型研究提供重要的理论基础。此模型包含 4 个不同的子模型，分别是：RABC0 模型、RABC1 模型、RABC2 模型、RABC3 模型。RBAC0 模型作为基本模型，主要定义了信息系统中最基本的访问控制需求；在 RBAC0 基础上添加了相应的层次化概念就是 RBAC1 模型，将约束引入 RBAC0 模型就是 RABC2 模型，而 RBAC3 模型没有再添加额外的概念，只是将前三种模型的所有概念整合起来，成为一个全新的模型[14]。图 1.1 是对这四个模型之间关系的一个简单描述。

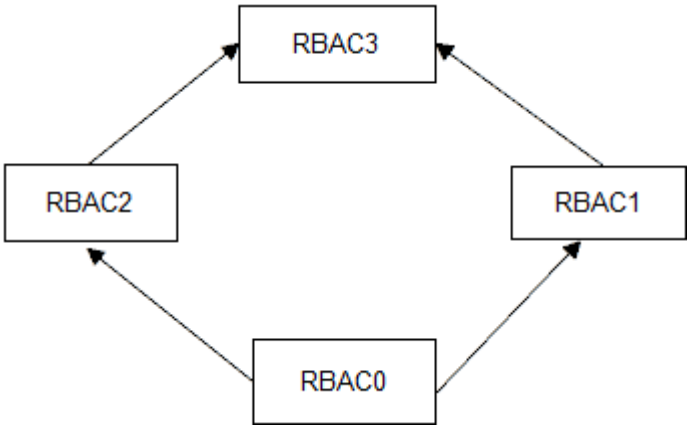

图 1.1 RBAC96 模型

网络及计算机上数据的不断的发展决定了无论是自主访问控制模型、 强制性访问控制模型还是基于角色的访问控制模型以及在RABC基础上引入属性概念的ABAC访问控制模型，都不可避免的逐渐显现出某一方面的不足[15]。以基于角色的访问控制模型为例，其设计访问控制策略在灵活性、网络环境的适应性、访问的安全性及动态化使用角色等方面逐渐表现出不足，已无法满足新型计算机系统的需求[16]。

被称作新一代访问控制模型的UCON，拥有上述传统访问控制模型所不具有的优势，例如它不仅针对封闭环境下的系统访问控制，还可以在开放的环境下对数据资源进行保护、不仅融合了以往传统模型中信任管理和数字版权管理中对数据库端和客户端数据的防护和使用控制，而且超越了两者的定义和范围，总的来说使用控制模型是对传统访问控制模型的继承与发展，并且包含了更高的使用要求和发展目标，进而成为当下访问控制模型领域的研究重点和热点，其最大的创新性在于提出属性的可变性以及访问控制过程的动态连续性，使得使用控制模型可以提供动态的、较为灵活的、 细粒度的访问控制

[17]。当然，使用控制模型也不尽完善，主要表现在缺乏理想的管理模型和委托[18]。

本文综合 RBAC 的优势，引进属性及角色概念对 UCON 模型进项扩展，提出一种动态授权模型 RAUCON，同时出了该模型具有时间约束的控制策略模型。并结合两个 RAUCON 模型的使用实例，验证其应用实现的可行性[19]。

1.2 本文的主要工作与章节安排

本文的研究借鉴了传统控制模型 RBAC 以及 UCON 的理论思想。主要研究工作有几下几点：

（1）首先较为详尽的描述了各类传统访问控制模型研究的背景及理论基础，分析了这几种模型的核心原理、特点及其相关的应用背景，比较了各类模型的优劣，最后提出本文的研究模型。

（2）针对 UCON 模型在委托方面的不足，进入角色概念，提出了一种带委托的授权模型 RAUCON，初步解决了委托授权的问题，对其核心子模型进行了较为详尽的形式化描述，并用两个简单的实例例子，验证该模型在实际应用中的可行性。

（3）研究了时限的特点，基于 RAUCON 模型，加入授权的时限约束，给出了一种比较简单的访问控制策略模型，并给出模型框架。本文的组织结构如下：

- 第一章 绪论。主要介绍了较为传统的几类访问控制模型的研究背景和动机，然后说明本文的主要工作和内容安排；
- 第二章 访问访问控制模型研究。较为简单的介绍了包括 DAC、MAC、RBAC 的在内的几类经典的访问控制模型的原理及特点。
- 第三章 针对 UCON 模型在委托角色及属性方面存有的缺陷，结合 RABC 和 UCON 模型，提出一种基于属于属性的使用控制模型 RAUCON，初步解决了 UCON 在委托角色及属性方面的问题。本章将详尽的描述 RAUCON 核心子模型体系，并用两个简答的实力验证该模型在实际应用中的可行性。
- 第四章 在 RAUCON 模型的基础上引入时限属性。描述了时间约束的相关特性，提出了具有时限性的 RAUCON 授权策略模型，并给出模型框架，分析域内及跨域访问的控制步骤。
- 第五章 总结与展望。总结了本文的研究工作，介绍后续的相关研究内容。

第二章 访问控制模型研究

2.1 访问控制技术

OSI 组织在 1989 年提出了 OSI 安全体系，访问控制模型作为其定义的五大安全服务之一，在维护网络体系的安全中具有不可替代的重要意义[20]。访问控制的实质是通过对系统资源进行限制访问，避免非法用户进入系统或合法用户因非法操作而对系统造成破坏及信息泄露，进而实现信息数据的安全。网络及计算机应用技术的不断研究和发展，也催生出许多较为主流的访问控制模型，他们分别是自主访问控制模型（DAC）、强制访问控制模型（MAC）、基于角色的访问控制模型（RABC）、基于属性的访问控制模型（ABAC）以及使用访问控制模型（UCON）[6]。本章将具体介绍以上集中模型[21]。

2.1.1 自主访问控制模型

自主访问控制模型（DAC，Discretionary Access Control），其自主性主要表现为主体能主动传递自身所有的客体访问权限给其他主体，同时又可以接受由其他主体赋予的操作权限，这使得授权操作具有较高的灵活性[22]。主体通过关联表实现对客体的访问等诸多操作，而关联表即可以基于主体又可以基于客体，两者的不同之处在于前者包含主体的访问策略、口令和规则、客体的信息，而后者则以客体为核心[23]。具体实现如下；首先要建一个访问控制矩阵，矩阵的行和列分别对应访问控制系统的主体和客体，主体对客体所具有的访问等操作权限则通过元素来描述。访问控制列表（ACL）[24]是自主访问控制模型较为典型的安全机制，如图 2.1 所示，表中的每个条目分别代表主体的身份以及对客体的访问操作权限[25]。

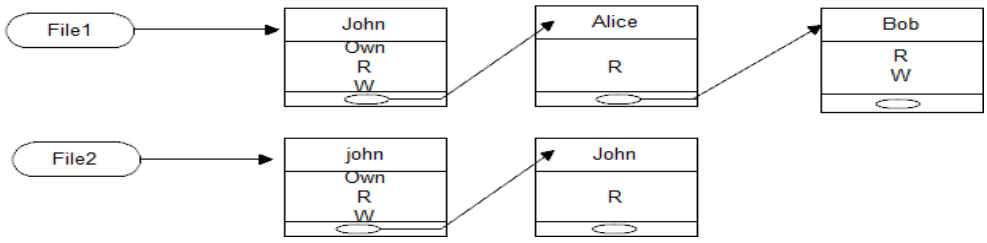

图 2.1 访问控制表（ACL）

上文提到了由于自主访问控制系统中的主体之间可以自主授权，访问控制具有非常高的灵活性，但也正因为这种授权特点使得主体访问权限过多，授权变得非常的困难，系统开销巨大[27]。此外在访问权的不断传递过程中，可能造成机密信息的意外泄露，造成较为严重的安全问题。

2.1.2 强制访问控制模型

强制访问控制模型（MAC，Mandatory Access Control），此模型的强制性主要表现在两个方面：一是其主体访问权限由授权机构统一设定，用户不能随意修改，二是强制控制数据的流向[29]，如图 2.2 所示，其规定数据只能从安全等级高的向安全等级低的流动（上读/下写）[30]。但也正因为这种严苛的数据流动方向控制，使得强制访问控制模型缺乏变通的灵活性，无法兼顾数据完整性和数据机密性，不便于管理。另外其基于无逆向潜信道的特性无法适应当今的计算机系统（难以消除逆向潜心道）[31]。

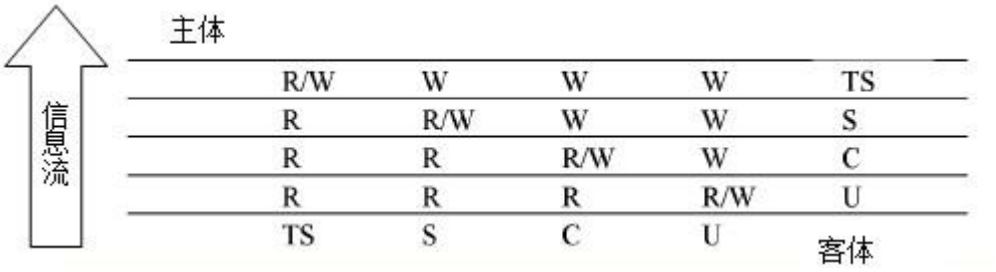

图 2.2　MAC 信息流控制

2.1.3 基于角色的访问控制模型（RBAC）

网络技术不断创新发展，必然推动访问控制模型的发展。为了更好地满足信息系统数据完整、机密性要求，弥补传统访问控制模型在诸多方面的不足，研究者们创新性的将角色这一概念进入到访问控制模型中，提出一种以角色为核心的访问控制模型（RBAC），后来又出现了较为经典的簇模型 RBAC96[31]。RBAC 与以往传统访问控制模型的不同之处在于用户访问等操作权限不由系统授予，而是包含在某一个特定的角色之中，系统只是定义角色与权限之前的关联，用户想要对某一客体的进行操作，只能通过被赋予包含相关权限的角色来实现，相当于在用户与权限之间建立一个中介，这样做的好处是系统组织中授权管理得到极大地简化，符合最小特权、全职分离及数据抽象安全

原则，不仅实现传统访问控制模型的思想，而且弥补自主与强制模型中的缺陷，具有更好的适应性[32]。

RBAC96 模型是基于角色的典型模型，也是后续多种基于角色扩展模型的基础。在其模型由 5 个基础元素组成：用户集合 U,角色集合 R，许可权集合 P，客体集合 O，回话集合 S[33]。各元素之间关系如图 2.3 所示。

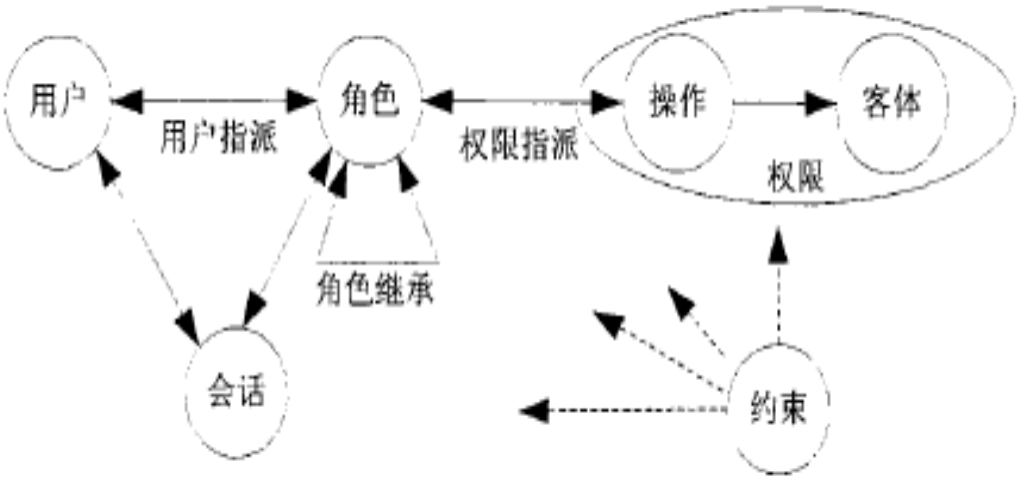

图 2.3　RBAC96 模型

用户集合表示对客体进行访问等操作的主体,角色集合其实是一类操作权限的集合，反应主体职责，客体集合指的是系统中可以被主体访问的数据等资源，权限集合是系统中数据等各类资源的访问操作许可集[34]。某个角色可以拥有多种访问权限，反过来某个权限也可同时被多个角色所拥有。会话集合中的会话是由用户激活相应角色建立的，是主体对系统资源进行访问的动态概念。用户与会话之间存在一个一对多的关系，即某个用户能够建立多个会话。RBAC96 模型由 4 个不同层次的子模型构成，分别为：基础子模型 RBAC0，只包含最基本的组成元素；层次模型 RBAC1，在第一个模型的基础上引入角色层次继承；限制模型 RBAC2，在包含　RBAC0 的基础上进入约束机制;综合模型 RBAC3；此模型没有再添加其它的概念，只是对前三种模型的整合[35]。图 2.4 显示了 96 模型的体系结构。RBAC 模型普遍应用于当下的诸多领域，例如操作系统、web 服务等。

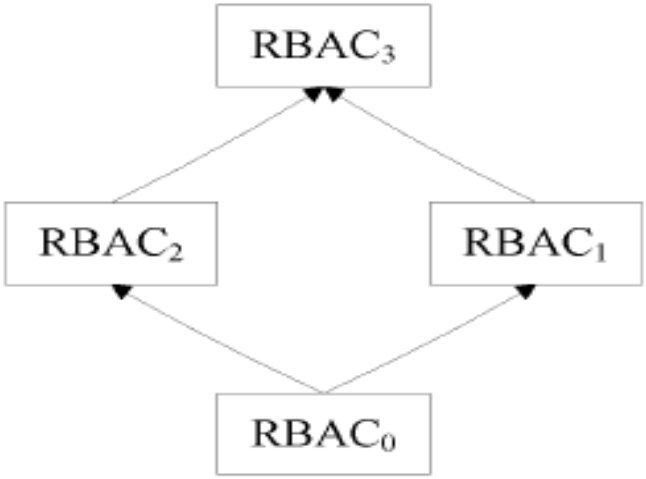

图 2.4 RBAC96 模型体系结构

1）RBAC0 模型

基础模型由用户集合（Users）、许可集合（Permission）、会话集合（Sessions）、角色集合（Roles）四个基础元素及用户角色分配（PA）、角色权限分配（UA）两个映射构成[。基本定义如下：

（1）U（用户集合），O（客体集合），P（许可集合），R（角色集合），S（会话集合）；

（2）$PA \subseteq P \times R$ 角色赋予用户（PA），是一个二元映射关系，角色与用户之间是一个多对多的关系；

（3）$UA \subseteq U \times R$ 权限授予角色（UA），是一个二元映射关系，权限与角色之间是一个多对多的关系；

（4）$user : S \rightarrow U$,会话映射到用户；

（5）$roles : S \rightarrow 2^R$，会话映射到角色集函数；

RBAC0 模型结构如图 2.5 所示：

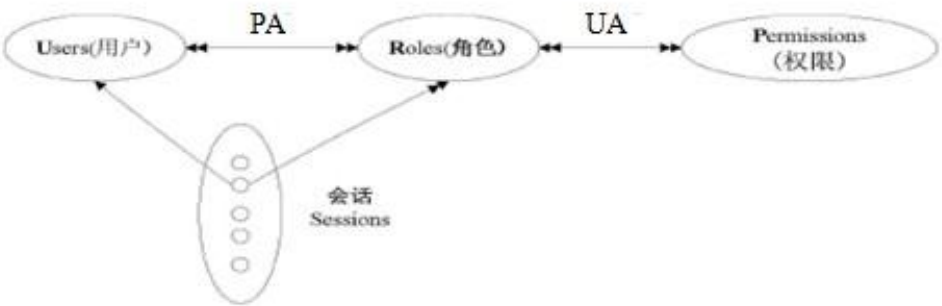

图 2.5　RBAC0 结构

2）RBAC1

此模型只是在上一个模型的基础上进入角色之间的一个层次结构，用来描述他们之间的继承关系（RH），好处是可以较好的反映一个组织内部的等级关系。

继承关系 $RH \subseteq R \times R$ 用来描述角色之间存在的继承关系；

$Roles: S \to 2^R$ 用户具有的全部权限；

3）RBAC2

此模型是在基础模型的基础上进入约束概念，用来判定各组成部分取值合法与否。约束包含多种类型，之中主要有职责分离、前提角色约束、互斥约束等。约束的作用一方面为了解决角色间潜在的冲突，比如互斥的角色不可赋予同一用户。

4）RBAC3

此模型没有在引进其他的概念，只是对前三种模型的整合统一，包含了前三种模型的全部内容，因而同时具有角色继承和条件约束。

2.1.4 基于属性的访问控制模型（ABAC）

此模型与先前集中介绍的模型一样包含主体集合、客体集合、许可集合这三个基本元素，不同之处在于这三个基本元素在此模型中都是用属性来表示，主体提供不同属性来判断是否有访问权限，授权操作也是由系统中属性判别式来控制。在访问控制中，策略执行的主体，客体资源和控制策略都是用资源和属性来描述。约束则用环境属性来表示，环境属性为动态属性，他是独立的不属于用户等其他基本元素[13]。ABAC 模型的结构如图 2.9 所示

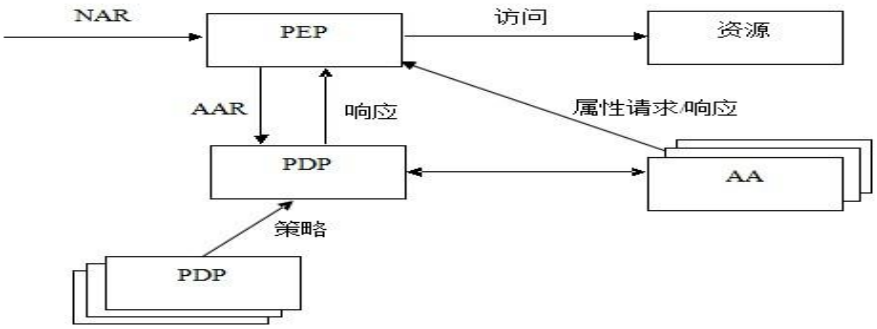

图 2.9 ABAC 模型示例

图中 NAR 为主体首次访问请求，AA 为属性权威，ARR 为属性控制的访问请求，

PDP、PAP 及 PEP 分别为控制策略的判定点、管理点和执行点。

2.1.5 使用控制模型 UCON

上述几种控制模型随着网络技术的不断发展，逐渐显现出它们的不足之处。这些不足只要表现在两个方面，一是通常关注的是一个封闭的系统，再者系统的访问控制都是基于主体、客体，都是被动型的安全模型，它们授权策略的静态性决定了用户从一开始被授权到访问操作结束，除非管理员主动撤销该用户权限，否则该用户被授予的权限能持续拥有，另外没有考虑到授权的上下文，授权系统存在一定隐患[37]。二是针对当下开放的分布式系统，传统控制模型已不能胜任。在封闭系统中，静态式授权的控制模型还在广泛应用着，但是网络技术的发展趋势是开放异构的分布式系统，需要更为复杂的动态授权机制。比如某个用户想从在某网站注册，他必须首先同意网站关于注册的各项要求并在规定的某个时间段注册方可成功，也就是说对于某种权限，用户必须完成一些任务或在特定条件下才能取得。这种授权策略就是基于主体的动作，环境条件等来完成，是当下访问控制不可或缺的决策因素[38]。这些授权因素还包括某些系统环境、上下文信息等等。专家们将这些因素大体分成三类，分别为职责、义务、条件。访问过程的连续性控制指的是授权决策能够在整个访问过程中有效，而不仅仅在请求提出时和访问客体前[39]。而所谓属性的易变性指的是主客体属性不仅仅在访问前对授权有影响，在访问的整个过程直至访问结束，主客体属性都可以发生变化，进而可以灵活控制访问过程中或接下来访问的授权决策，访问控制的授权决策受到属性的影响，系统在必要时能够实时收回不满足授权决策的用户权限[40]。2003 年，R.Sandhu 与 J.Parkt 给出使用控制核心模型的完整定义，也成为 ABC 模型[1]41，如图 2.10 所示：

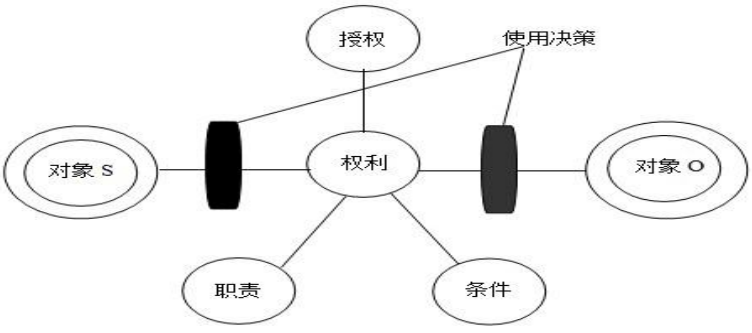

图 2.10　使用控制模型框架

在此模型中，使用决策受三个因素共同影响，分别是授权规则、条件、义务[41]。使用控制模型能够实现传统控制模型的授权等控制策略，同时满足信任和数字版权关系的要求，实现多种复杂授权策略，不仅如此，使用控制还涵盖了信息系统对保护隐私和安全问题的要求[42]。

可变的主客体属性以及连续的使用控制两大特性，是 UCON 区别于其他访问控制模型的关键。如图 2.11 所示。当下的访问控制，需要能够对系统中的客体资源进行持续的访问，又可以在必要时撤销用户的某项操作权限，这需要对主体的整个访问过程进行实时连续的监控。UCON 的连续性控制作用于整个访问过程[43]。并且涵盖了包括控制技术、数字版权和信任管理等多项内容[44]。

图 2.11 使用控制模型的可变性与连续型

UCON 上述三个决定因素以及属性更新时间段的不同，可以划分为 16 种子模型，如表 2.1 所示。通过组合表中的各种子模型，UCON 可以实现多种不同的控制机制[45]。

表 2.1 ABC 模型矩阵

	0 （不可变化）	1 （使用前更新）	2 （使用中更新）	3 （使用后更新）
preA	Y	Y	N	Y
onA	Y	Y	Y	Y
preB	Y	Y	N	Y
onB	Y	Y	Y	Y
preC	Y	N	N	N
onC	Y	N	N	N

2.2 UCON 的不足

虽然 UCON 作为当今较为优秀的控制模型被本领域的研究者作为热点，但多为停留在理论概念上的研究，该模型仍有很多待研究者们去完善的不足[46]，比如：

（1）其模型框架不够丰富。现在为止的 UCON 很大程度上仍然停留在统一传统控制、信任等管理的定义和方法上，模型还有一定的抽象性，有进一步具体和完善的研究需要。

（2）没有较为优良的管理模型，使得 UCON 模型的属性和授权的缺乏有效管理，这也是将来对 UCON 模型研究的一个重要的内容[47]。

（3） 在 UCON 模型表达能力较强，应用前景广的前提下，其应用应用模型的研究同样是接下来刻不容缓的一项研究关键。

（4）虽然部分研究者虽提出了 UCON 委托授权问题的一些解决办法,但就委托属性、委托策略及委托撤销管理的研究较少[49]。

2.3 本章小结

本章比较详细的介绍了包括自主、强制、基于角色的访问控制模型在内的多种控制模型，阐述了各模型的核心思想，比较了它们各自的优缺点以及提出背景，例如 DAC 和 MAC 在全局访问控制方面的不足、连续控制系统以及管理授权灵活性方面有所欠缺等。

第三章 基于属性 RBAC 的使用控制模型研究

本章节将根据 UCON 在委托授权方面存在的不足[50]，以及 ABAC[27]和 RBAC 于策略模型的表述，结合静态、动态属性，建立一种基于属性 RBAC 的使用控制模型（Rose Attribute Based Usage Control）RAUCON。建立角色与属性之间的对应关系，以及用户、属性等基础元素之间的授权关系，给出模型授权决策模型和具体实现例子。

3.1 基于属性 RBAC 的使用控制模型（RAUCON）

3.1.1 模型定义与性质

基于属性 RBAC 的使用控制模型思想是用属性来描述 RBAC 中的角色，将其分为静态属性（例如地址、住址、名称等较为固定的性质，这些不同的属性组合成不同的类角色）和动态属性（例如日期、月份等易于变动的性质以及委托属性）然后引入到使用控制模型中，解决 UCON 权限、委托授权以及权限管理方面的缺陷。该模型是以 UCON 模型为基础的，基于扩展属性的访问控制方法，融入扩展属性等要素，实现动态管理和分配用户权限。RAUCON 的使用决策有四个因素决定，分别为授权、条件、义务和委托，他们组成的决策模块同时对用户进行动态的管理。RAUCON 模型框架如图 3.1 所示：

如图 3.1 可以看出，模型 RAUCON 主要由主体（出自 UCON）、客体（出自 UCON）授权（出自 UCON）、条件（出自 UCON）、义务（出自 UCON）、角色（引入概念）等六个基本元素构成。

（1）主体是 RAUCON 中通过被赋予权限而对客体进行相应操作的实体，记为 S。

（2）客体是系统中能够被用户进行访问等操作的资源的集合，记为 O。

（3）权限是一组操作许可的集合，表示主体主体能够对客体施加的一些具体操作，记为 P。

（4）角色是 RABC 中的引入的概念，在 RAUCON 中角色通分为普通角色和委托角色，普通角色通过属性进行描述，而委托角色是经过某次委托授权而获得的角色，记为 R。

（5）授权是一组判断操作，根据特定的授权策略规则，同时考虑主客体属性的变化，在整个访问过程中进行前、中、后的授权判断。

（6）义务和条件来自 UCON，由于概念一样在这里就不再具体叙述，记为 C。

（7）委托的基本概念是系统中的活跃实体将本身拥有的某种权限授予其他实体，以便使其代替自己完成对客体资源的某种操作，记为 D。例如；主体 A 受到来自主体 B 的委托属性，那么主体 A 便将委托属性融入到自己的动态属性当中。

RAUCON 由授权、义务、条件、以及委托这四个因素来决定其使用决策，这四个因素也就构成了 $RAUCON_{0123}$ 模型。

3.2 RAUCON 模型体系

由于 PAUCON 模型的使用策略由四个主要基本元素构成，再加上访问的整个过程中主、客体以及环境属性是否存在变化，可以将 PAUCON 分成 18 个子模型。图 3.2 是由授权等四个决定元素与相关属性是否发生改变及变化时间等限定条件组合而成的 RAUCON 子模型矩阵。

表 3.1 PAUCON 模型矩阵

	0 (不可变化)	1 （使用前更新）	2 （使用中更新）	3 （使用后更新）
preA	Y	Y	N	Y
onA	Y	Y	Y	Y
preB	Y	Y	N	Y
onB	Y	Y	Y	Y
preC	Y	N	N	N
onC	Y	N	N	N
preD	N	Y	N	N
onD	N	N	Y	N

此表的某些表示与 UCON 一致，比如数字 0 表明在访问过程中没有不存在属性更新变化 ，数字 1 表示存在更新变化，且发生在前。数字 2 同样表示有属性变化出现，不同的是在使用前完成属性的更新。数字 3 代表使用后出现的属性更新。字母 Y 表示存在模型，N 表示不存在。图 3.2 描述出了 RAUCON 个子模型之间的逻辑关系。大体可以分为 4

种情况，本章节下来的篇幅将对形式化描述其部分子模型。

3.2.1 RAUCONO 模型

3.2.1.1 $RAUCON_{pre0}$

此模型为预授权模型，由子模型 $RAUCON_{pre00}$，表示在使用过程中属性不更新、子模型 $PAUCON_{pre01}$，表示属性在使用前出现更新、子模型 $RAUCON_{pre02}$，表示使用后属性出现更新情况组成。

定义 3.1 $PAUCON_{pre01}$ 模型

（1）主体 S、客体 0、主体属性 SA、客体属性 OA、角色 R、权限 P、权限预授函数 PreA；

（2）$contain(r,p)\rightarrow\{true, flase\}$;判别权限 P 是否授予了角色 R，true 为授予，未授予为 false;

（3）$grant(r.s)\rightarrow\{true, flase\}$;判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

（4）$allowed(s,o,p)\Leftarrow pre0(p,Sa,Oa)\wedge contain(p,r)\wedge grant(r,s)$。此函数表示主体获得访问权限的条件是 1.主、客体属性，以及主体想要取得的操作权限满足权限预授函数。2.主体需拥有已包含了相应访问权限 P 的角色 R。

定义 3.2 $PAUCON_{pre01}$ 模型

此模型为属性使用前更新模型，其中主、客体及其属性等都与 $PAUCON_{pre00}$ 模型大体一致，仅仅是在上一个模型的基础上添加了一个属性更新的操作。

（1）$BFUpdate(Sa)$ 主体属性使用前更新函数。

（2）$BFUpdate)(Oa)$ 客体属性的使用前更新函数。

定义 3.3 $RAUCON_{pre02}$ 模型

此模型与 $RAUCON_{pre01}$ 模型很相似，仅仅是将使用前的属性更新改为使用后的属性更新。

（1）$AfUpdate(Sa)$，主体属性的使用后更新函数。

（2）$AfUpdate(Oa)$，客体属性的使用后更新函数。

3.2.1.2 $RAUCON_{on0}$

与传统$UCON_{onA}$模型含义基本相同，在使用前是不需要进行任何判别操作的，但是在主体进行访问的过程中，要求持续或在固定的时间段对授予的权限进行判决，当主体或客体及其属性等因素不再符合规定的授权规则时，主体的权限会立刻被撤销。他有四个子模型，分别为全程属性不更新子模型:$RAUCON_{on00}$,属性更新发生在使用前子模型：$RAUCON_{on01}$，$RAUCON_{on02}$为属性更新发生在使用中的子模型以及属性发生在使用后的子模型$RAUCON_{on03}$.下面给出这几个此模型的定义：

定义 3.4 $RAUCON_{on00}$

（1）模型中的基本组成元素有主体 S，客体 0，主属性 Sa,客属性 0a，角色 r,权限 p，角色授予判定函数 grant(r,s),权限包含判定函数 contain(p,r)。使用中授权函数 onA 。$\text{authority}(s,p,o) \equiv \text{true}$，访问前不需要进行任何的授权判决操作。

（2）$revoke(s,p,o) \Leftarrow \neg onA(Sa,Oa,p) \vee \neg contain(p,r) \vee grant(r,s)$，主体对于可以的访问权限将在下列情况被撤销：1.授权函数不再成立。2.主体没有被授予包含相应权限的角色。

定义 3.5 $RAUCON_{on01}$

（1）模型中的基本组成元素有主体 S，客体 0，主属性 Sa,客属性 0a，角色 r,权限 p，角色授予判定函数 grant(r,s),权限包含判定函数 contain(p,r)。使用中授权函数 onA 。$\text{authority}(s,p,o) \equiv \text{true}$，访问前不需要进行任何的授权判决操作。

（2）$BFUpdate(Sa)$,主体属性使用前更新函数。

（3）$BFUpdate(Oa)$,客体属性的使用前更新函数。

定义 3.6 $RAUCON_{on02}$模型

（1）模型中的基本组成元素有主体 S，客体 0，主属性 Sa,客属性 0a，角色 r,权限 p，角色授予判定函数 grant(r,s),权限包含判定函数 contain(p,r)。使用中授权函数 onA 。$\text{authority}(s,p,o) \equiv \text{true}$，访问前不需要进行任何的授权判决操作。

（2）$underUpdate(Sa)$,主体属性的使用中更新操作函数。

（3）$underUpdate(Oa)$，客体属性的使用中更新操作函数。

定义 3.7 $RAUCON_{on03}$模型

（1）模型中的基本组成元素有主体 S，客体 0，主属性 Sa,客属性 0a，角色 r,权限

p，角色授予判定函数 grant(r,s),权限包含判定函数 contain(p,r)。使用中授权函数 onA 。 $authority(s,p,o) \equiv true$，访问前不需要进行任何的授权判决操作。

（2） $AfUpdate(Sa)$，主体属性的使用后更新函数。

（3） $AfUpdate(Oa)$，客体属性的使用后更新函数。

3.2.2 RAUCON1 模型

3.2.2.1 $RAUCON_{pre1}$

模型为前义务，也即是说此模型的授权决策包含有义务这一谓词，义务作为决策因素参与用户请求的判断。比如，学生在登陆学校的学生管理系统时，必须输入正确的学号、密码的信息，才能成功进入。我们在网站注册会员时，往往必须先阅读该网站的注册要求，才能进行下一步的注册操作。当下视频网站提供在线视频时其中附加的广告，等都是所谓的义务。因为模型规定用户在进行某种使用操作之前，往往要额外完成一些系统附带或要求的任务，使得该模型一般分两步执行。另外，义务在很多情况下会对属性产生影响，使得属性发生变化。据此，前义务模型可分为全程无更新模型 $PAUCON_{pre10}$、前更新模型 $RAUCON_{pre11}$ 模型和后更新属性模型 $RAUCON_{pre12}$ 模型。

定义 3.8 $RAUCON_{pre10}$

(1) $grent(r,s)$， $contain(p,r)$，主体 S、客体 0、主体属性 SA、客体属性 OA。

(2) $contain(r,p) \rightarrow \{true, flase\}$;判别权限 P 是否授予了角色 R，true 为授予，未授予为 false;

(3) $grant(r.s) \rightarrow \{true, flase\}$;判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

(4) $authority(s,o,p) \Leftarrow pre0(p,Sa,Oa) \wedge contain(p,r) \wedge grant(r,s)$。此函数表示主体获得访问权限的条件是 1.主、客体属性，以及主体想要取得的操作权限满足权限预授函数。2.主体需拥有已包含了相应访问权限 P 的角色 R。

(5)OBS 义务的主体，OBO 义务的客体，OBAT 义务的动作;

（6） $preOBL \subseteq OBS \times OBO \times OBAT$，描述了主体所需要满足的所有预先义务。 $preOBL(obsi,oboi,obati)$ 代表着其中的某一个义务元素。

（7） $getpreOBL: S \times O \times R \to 2^{preOBL}$，此公式通过判断主、客体以及所求权限匹配相应的预先义务集合，实现主、客体及其所求权限与义务主、客体及动作的映射，从而得到主体在获得所求权限前需要履行的义务的集合。至此就完成了预先义务模型的第一步。

（8） $prefulfill\,ed: OBS \times OBO \times OBAT \to \{true, flase\}$, 此函数主要用来判断上一步所得出的义务是否被履行，$prefulfill\,ed(obsi, oboi, obati)$ 描述义务主体针对义务客体是否满足特定义务动作的要求，而这里的义务主、客体及其义务动作必须是 *prefulfilled* 所得义务集合中的一个。

（9） *preB* 使用来判别 *getpreOBL* 得到的义务集合是否能够被履行的预先义务函数。 $preB(s, o, p) = \wedge_{(obsi, oboi, obati) \in getpreOBL(s,o,p)} prefulfilled(obsi, oboi, obi)$ 此函数判别所需义务是否已被履行。可以看出，如果没有义务需要被履行，也就是 $getpreOBL(s, o, p) = \Phi$，那么 $preB(s, o, p) = true$ 。

（10）$allowed(s, o, p) \Leftarrow preB(s, o, p) \wedge contain(p, r) \wedge grent(r, s)$, 此函数值为真的条件即请求主体取得权限的条件是：1. 所有需要被履行的义务元素履行完毕。2. 请求主体拥有包含特定权限的角色。

定义 3.9 $RAUCON_{pre11}$ 模型

此模型仅仅是在上一个模型的基础上添加了属性更新操作。

(1)主体 S、客体 0、主体属性 SA、客体属性 OA、角色 R、权限 P、权限预授函数 Pre0;

(2) $contain(r, p) \to \{true, flase\}$; 判别权限 P 是否授予了角色 R，true 为授予，未授予为 false;

(3) $grant(r.s) \to \{true, flase\}$; 判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

(4) $\text{authority(s,o,p)} \Leftarrow \text{pre0(p,Sa,Oa)} \wedge \text{contain(p,r)} \wedge \text{grant(r,s)}$ 。此函数表示主体获得访问权限的条件是 1. 主、客体属性，以及主体想要取得的操作权限满足权限预授函数。2. 主体需拥有已包含了相应访问权限 P 的角色 R。

(5)OBS 义务的主体，OBO 义务的客体，OBAT 义务的动作;

（6） $preOBL \subseteq OBS \times OBO \times OBAT$ ，描述了主体所需要满足的所有预先义务。

$preOBL(obsi,oboi,obati)$ 也就代表着其中的某一个义务元素。

（7） $getpreOBL:S\times O\times R\to 2^{preOBL}$，此公式通过判断主、客体以及所求权限来匹配相应的预先义务集合，实现主、客体及其所求权限与义务主、客体及动作的映射，从而得到主体在获得所求权限前，需要履行的全部义务集合。至此就完成了预先义务模型的第一步。

（8）$prefulfilled:OBS\times OBO\times OBAT\to\{true, flase\}$，此函数主要用来判断上一步所得出的义务是否被履行，$prefulfill\,ed(obsi,oboi,obati)$ 作用是判断义务主体针对义务客体，是否满足特定义务动作的要求，而这里的义务主、客体及其义务动作必须是 $prefulfill\,ed$ 所得义务集合中的一个。

（9） $preB$ 使用来判别 $getpreOBL$ 得到的义务集合是否能够被履行的预先义务函数。 $preB(s,o,p)=\wedge_{(obsi,oboi,obati)\in getpreOBL(s,o,p)}prefulfilled(obsi,oboi,obi)$ 此函数判别所需义务是否已被履行。很容易可以得出如果没有义务需要被履行，也就是 $getpreOBL(s,o,p)=\Phi$，那么 $preB(s,o,p)=true$。

（10） $allowed(s,o,p)\Leftarrow preB(s,o,p)\wedge contain(p,r)\wedge grent(r,s)$ 此函数值为真的条件即请求主体取得权限的条件是：1.所有需要被履行的义务元素履行完毕。2.请求主体拥有包含特定权限的角色。

（11） $BfUpdate(Sa)$，主属性的使用前更新函数。

（12） $BfUpdate(Oa)$，客属性的使用前更新操作函数。

定义 3.10 $RAUCON_{pre13}$ 模型

此模型是拥有属性使用后更新操作的模型。

(1) $grent(r,s)$， $contain(p,r)$，主体 S、客体 0、主体属性 SA、客体属性 OA。

(2) $contain(r,p)\to\{true, flase\}$;判别权限 P 是否授予了角色 R，true 为授予，未授予为 false;

(3) $grant(r.s)\to\{true, flase\}$;判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

(4) $\text{authority(s,o,p)}\Leftarrow\text{pre0(p,Sa,Oa)}\wedge\text{contain(p,r)}\wedge\text{grant(r,s)}$。此函数表示主体获得访问权限的条件是 1.主、客体属性，以及主体想要取得的操作权限满足权限预授函数。2.主体需拥有已包含了相应访问权限 P 的角色 R。

(5)OBS 义务的主体，OBO 义务的客体，OBAT 义务的动作；

（6） $preOBL \subseteq OBS \times OBO \times OBAT$，描述了主体所需要满足的所有预先义务，而 $preOBL(obsi, oboi, obati)$ 也就代表着其中的某一个义务元素。

（7） $getpreOBL : S \times O \times R \to 2^{preOBL}$，此公式通过判断主、客体以及所求权限匹配相应的预先义务集合，实现主、客体及其所求权限与义务主、客体及动作的映射，从而得到主体在获得所求权限前需要履行的义务的集合。至此就完成了预先义务模型的第一步。

（8）$prefulfilled : OBS \times OBO \times OBAT \to \{true, flase\}$，此函数主要用来判断上一步所得出的义务是否被履行， $prefulfilled(obsi, oboi, obati)$ 是描述义务主体针对义务客体，是否满足特定义务动作的要求，而这里的义务主、客体及其义务动作必须是 $prefulfilled$ 所得义务集合中的一个。

（9） $preB$ 使用来判别 $getpreOBL$ 得到的义务集合是否能够被履行的预先义务函数。 $preB(s, o, p) = \wedge_{(obsi, oboi, obati) \in getpreOBL(s, o, p)} prefulfilled(obsi, oboi, obi)$，此函数判别所需义务是否已被履行。很容易可以得出如果没有义务需要被履行，也就是 $getpreOBL(s, o, p) = \Phi$，那么 $preB(s, o, p) = true$。

（10） $allowed(s, o, p) \Leftarrow preB(s, o, p) \wedge contain(p, r) \wedge grent(r, s)$，此函数值为真的条件即请求主体取得权限的条件是：1.所有需要被履行的义务元素履行完毕。2.请求主体拥有包含特定权限的角色。

（11） $AfUpdate(Oa)$，客体属性使用后的更新操作函数。

（12） $AfUpdate(Sa)$，主体属性使用后的更新操作函数。

3.2.2.2 $RAUCON_{on1}$

此模型称为使用中的义务模型，即主体在再行使权力的同时，需要在特定时间段或者整个过程持续的履行某项义务。整个过程需要进入一个参数 T 做为执行中义务集合的一个元素。T 可以表示某个特定的时间段或者是某个特定的时刻。例如，用户在某个网站进行在线阅读时必须每隔 20 分钟要进行一个该网站的推广操作或在阅读的同时必须一直开着该网站的某个播放广告的小窗口，而参数 T 就用来记录这些义务操作的时间集合。在此模型中，基于属性的变化，可以将该模型划分为 4 个相应的子模型：$RAUCON_{on10}$、

$RAUCON_{on11}$、$RAUCON_{on12}$、$RAUCON_{on13}$。

定义 3.11 $RAUCON_{on10}$模型

此模型表示在使用过程中不发生属性更新的操作。其描述如下：

(1) $grent(r,s)$，$contain(p,r)$，主体 S、客体 0、主体属性 Sa、客体属性 Oa，r,p。

(2) $contain(r,p)\rightarrow\{true, flase\}$;判别权限 P 是否授予了角色 R，true 为授予，未授予为 false;

(3) $grant(r.s)\rightarrow\{true, flase\}$;判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

(4) $\text{authority(s,o,p)}\Leftarrow\text{pre0(p,Sa,Oa)}\wedge\text{contain(p,r)}\wedge\text{grant(r,s)}$。此函数表示主体获得访问权限的条件是 1.主、客体属性，以及主体想要取得的操作权限满足权限预授函数。2.主体需拥有已包含了相应访问权限 P 的角色 R。

(5)OBS 义务的主体，OBO 义务的客体，OBAT 义务的动作。

（6） $onOBL\subseteq OBS\times OBO\times OBAT\times T$，描述了主体所需要满足的所有预先义务，而 $preOBL(obsi,oboi,obati,ti)$ 也就代表着其中的某一个义务元素。

（7） $getonOBL:S\times O\times P\rightarrow 2^{onOBL}$，此公式通过主、客体以及所求权限匹配相应的预先义务集合，实现主、客体及其所求权限与义务主、客体及动作的映射，从而得到主体在获得所求权限前需要履行的义务的集合。

（8）$onfulfilled:OBS\times OBO\times OBAT\times T\rightarrow\{true, flase\}$,此函数主要用来判断上一步所得出的义务是否被按时履行，$onfulfilled(obsi,oboi,obati)$ 是描述义务主体针对义务客体是否满足特定义务动作的要求，而这里的义务主、客体及其义务动作必须是 $getonOBL(s,o,p)$ 所得义务集合中的一个。

（9） onB 用来判别 $getonOBL()$ 得到的义务集合是否能够被履行的预先义务函数。$onB(s,o,p)=\wedge_{(obsi,oboi,obati)\in getonOBL(s,o,p)}onfulfilled(obsi,oboi,obi)$，此函数判别所需义务是否已被履行。很容易可以得出如果没有义务需要被履行，也就是 $getonOBL(s,o,p)=\Phi$，那么 $onB(s,o,p)=true$。

（10） $allowed(s,o,p)\equiv true$，使用前无需进行任何义务判断。

（11） $revoke(s,o,p)\Leftarrow\neg onB(s,o,p)\vee\neg grant(r,s)\vee\neg contain(p,r)$，在使用过程中主体权限会被系统收回的两个条件分别为：1.过程中所需履行的义务没有按时执行。或者

主体不再拥有包含特定权限的脚色。

定义 3.12 $RAUCON_{onli}$ 模型

此模型具有属性更新的操作。其逻辑描述为：

(1) $grent(r,s)$，$contain(p,r)$，主体 S、客体 0、主体属性 Sa、客体属性 Oa，r,p。

(2) $contain(r,p)\rightarrow\{true, flase\}$;判别权限 P 是否授予了角色 R，true 为授予，未授予为 false;

(3) $grant(r.s)\rightarrow\{true, flase\}$;判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

(4) $\text{authority(s,o,p)}\Leftarrow\text{pre0(p,Sa,Oa)}\wedge\text{contain(p,r)}\wedge\text{grant(r,s)}$。此函数表示主体获得访问权限的条件是 1.主、客体属性，以及主体想要取得的操作权限满足权限预授函数。2.主体需拥有已包含了相应访问权限 P 的角色 R。

(5)OBS 义务的主体，OBO 义务的客体，OBAT 义务的动作。

（6） $onOBL\subseteq OBS\times OBO\times OBAT\times T$，描述了主体所需要满足的所有预先义务。$preOBL(obsi,oboi,obati,ti)$ 也就代表着其中的某一个义务元素。

（7） $getonOBL:S\times O\times P\rightarrow 2^{onOBL}$，此公式通过主、客体以及所求权限匹配相应的预先义务集合，实现主、客体及其所求权限与义务主、客体及动作的映射，从而得到主体在获得所求权限前需要履行的义务的集合。

（8）$onfulfilled:OBS\times OBO\times OBAT\times T\rightarrow\{true, flase\}$,此函数主要用来判断上一步所得出的义务是否被按时履行，$onfulfilled(obsi,oboi,obati)$ 是描述义务主体针对义务客体是否满足特定义务动作的要求，而这里的义务主、客体及其义务动作必须是 $getonOBL(s,o,p)$ 所得义务集合中的一个。

（9） onB 使用来判别 $getonOBL()$ 得到的义务集合是否能够被履行的预先义务函数。$onB(s,o,p)=\wedge_{(obsi,oboi,obati)\in getonOBL(s,o,p)}onfulfilled(obsi,oboi,obi)$，此函数判别所需义务是否已被履行。很容易可以知道如果没有义务需要被履行，也就是 $getonOBL(s,o,p)=\Phi$，那么 $onB(s,o,p)=true$。

（10） $allowed(s,o,p)\equiv true$，使用前无需进行任何义务判断。

（11）$revoke(s,o,p)\Leftarrow\neg onB(s,o,p)\vee\neg grant(r,s)\vee\neg contain(p,r)$，在使用过程中主体权限会被系统收回的两个条件分别为：1.过程中所需履行的义务没有按时执行。或

者主体不再拥有包含特定权限的脚色。

（12） *BfUpdate*(*Sa*)，主属性的使用前更新函数。

（13） *BfUpdate*(*Oa*)，客属性的使用前更新操作函数。

定义 3.13 $RAUCON_{on12}$ 模型

此模型同样具有属性的更新操作。与 $RAUCON_{on10}$ 模型的定义大体一致，仅仅添加了两个属性更新操作的函数。

（1） *onUPdate*(*Sa*)，主属性在使用中的更新函数。

（2） *onUpdate*(*Oa*)，客属性在使用中的更新函数。

定义 3.14 $RAUCON_{on13}$ 模型

此模型同样具有属性的更新。与 $RAUCON_{on10}$ 模型的定义大体一致，仅仅添加了两个属性更新操作的函数。

（1） *AfUpdate*(*Sa*)，主属性的使用后更新函数。

（2） *AfUpdate*(*Oa*)，客属性的使用后更新操作函数。

3.2.3 RAUCON2 模型

此模型为包含条件约束的模型。这里的条件含义以传统 UCON 中大体一致，都是在授权于义务之外，还将条件作为一个决策因素包含在定以内。这里的条件有很多方面的指向，比如某种环境约束或者系统约束等，这些条件大多数情形下虽然不与主、客体及其属性产生交集，却能规范整个交互环境及访问请求产生影响。

3.2.3.1 $RAUCON_{pre2}$

此模型包含预先条件概念。将条件因素添加入使用决策之中，产生更科学合理的使用前授权模型对象。在此模型中，用户想要获得所求权限，必须执行两步，分别为：1.满足所有系统预先要求的条件元素。2.与请求对应的所有条件元素必须都被正确的执行。在条件模型中，属性是全程无变化的，所以此模型只包含一个子模型。

定义 3.15 $RAUCON_{pre20}$ 模型

此模型的逻辑描述如下：

（1）s,o,r,p,Sa,Oa,grant(r,s),contain(p,r)。这些元素的含义与前几个模型中大

体一致，在这里不再赘述。

（2） *preCL* 为主体获取所求权限应满足的预先条件集。

（3） $getpreCL: S \times O \times P \to 2^{preCL}$，请求过程中，根据主、客体及其属性和主体所求权限来确定需要满足的预先条件。这是此模型执行的第一步。

（4） $preConChecked: preCL \to \{true, false\}$，用来判别 *getpreCL* 得到的条件是否已被满足。 $preConChecked(preCL)$ 用来检测某个具体条件 *preCLi* 满足与否，当然这个元素必须是由 *getpreCL*() 得到的。

（5） $preC(s,o,p) = \wedge_{preCLi \in getpreCL(s,o,p)} preConChecked(preCLi)$, 判断得到的预先条件元素是否能够被满足，最后将其结果进行与运算。

（6） $allowed(s,o,p) \Leftarrow preC(s,o,p) \wedge grant(r,s) \wedge contain(p,r)$，表示主体能够获得所求权限的条件为：1. 满足全部系统要求预先义务元素。2. 主体拥有包含特定权限的角色。

3.2.3.2 $RAUCON_{on2}$

此为使用过程中的条件模型，即某些条件约束作用于使用的过程。基于属性的不可变性，此模型同样仅有一个子模型 $RAUCON_{on20}$。

定义 3.16 $RAUCON_{on20}$ 逻辑描述如下：

（1）s,o,r,p,Sa,Oa,grant(r,s),contain(p,r)。这些元素的含义与前几个模型中大体一致，在这里不再赘述。

（2） *onCL* 为主体获取所求权限应满足的使用中条件集。

（3） $getonCL: S \times O \times P \to 2^{onCL}$，请求过程中，根据主、客体及其属性和主体所求权限来确定需要满足的预先条件。这是此模型执行的第一步。

（4） $onConChecked: onCL \to \{true, false\}$，用来判别 *getonCL* 的到的条件是否已被满足。 $onConChecked(onCL)$ 用来检测某个具体条件 *onCLi* 满足与否，当然这个元素必须是由 *getonCL*() 得到的。

（5） $onC(s,o,p) = \wedge_{onCLi \in getonCL(s,o,p)} onConChecked(onCLi)$, 判断得到的使用中条件元素是否能够被满足，最后将其结果进行与运算。

（6） $allowed(s,o,p) \equiv true$，此模型访问前不许要任何条件检测操作。

（7） $revoke(s,o,r,p) \Leftarrow \neg onC(s,o,p) \vee \neg grant(s,r) \vee contain(p,r)$。表示主体所求权

限被撤销的条件为：1.条件在主体的访问过程中不再被满足。2.主体不再拥有包含特定权限的角色。

3.2.4 RAUCON3 模型

此模型为委托模型，是将委托因素引入授权决策得出的细粒度模型，包括属性以及角色的委托。委托在大多数情况下会导致属性发生改变。

3.2.4.1 $RAUCON_{pre3}$

定义 3.17 $RAUCON_{pre31}$ 模型的逻辑描述如下:

（1）主体 S、客体 O、主体属性 SA、客体属性 OA、角色 R、权限 P、委托属性 e，委托角色 re。$contain(r,p)\rightarrow\{true, flase\}$;判别权限 P 是否授予了角色 R，true 为授予，未授予为 false； $grant(r.s)\rightarrow\{true, flase\}$;判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

（2） $preEA(Sa,e,Oa,p)$,用来描述接受来自其他主体的属性委托后，所求权限是否被授予。

（3）$preER(r,re,Oa,p)$,用来描述接受来自其他主体的角色委托后，所求权限是否被授予。

（4）$preE \Leftarrow preEa\times preEr$,在委托模型中，预先授权判决的委托因素由两部分组成，分别是委托属性以及角色。

（5） $allowed(s,o,p)\Leftarrow preE\wedge grant(r,s)\wedge contain(p,r)$,表示在委托模型中，主体能够获得所求权限的条件为：1.主、客属性及主体所求权限满足相应的预委托函数。2.主体拥有包含特想权限的角色。

(6) $BfUpdate(e)$,表示委托属性后主属性的更新函数。

(7 $BfUpdate(re)$,表示委托角色后主属性的更新操作函数。

(8) $BfUpdate(Oa)$,客属性的使用前属性更新函数。

3.2.4.2 $RAUCON_{on3}$

此模型为使用中委托模型，即在主体访问的过程中进行特定时间间隔或时刻的委托

判断，如果主体正在进行的操作已不再满足相应的委托时，系统会收回主体的权限。次模型也仅有一个具体模型。

定义 3.18

$RAUCON_{on32}$逻辑描述如下：

（1）主体 S、客体 O、主体属性 SA、客体属性 OA、角色 R、权限 P、委托属性 e，委托角色 re。$contain(r,p)\rightarrow\{true, flase\}$;判别权限 P 是否授予了角色 R，true 为授予，未授予为 false；$grant(r.s)\rightarrow\{true, flase\}$;判别主体 S 是否拥有角色 R，true 表示拥有，如果还没授予就为 false;

（2）$onEA(Sa,e,Oa,p)$,用来描述执行过程中接受来自其他主体的属性委托后，所求权限是否被授予。

（3）$onER(r,re,Oa,p)$,用来描述执行过程中接受来自其他主体的角色委托后，所求权限是否被授予。

（4）$onE\Leftarrow onEa\times onEr$,在委托模型中,预先授权判决的委托因素由两部分组成，分别是委托属性以及角色。

（5）$allowed(s,o,p)\Leftarrow onE\wedge grant(r,s)\wedge contain(p,r)$,表示在委托模型中,主体能够获得所求权限的条件为:1.主、客属性及主体所求权限满足相应的执行中委托函数。2.主体拥有包含特想权限的角色。

（6）$onUpdate(e)$,表示委托属性后主属性的更新函数。

（7）$onUpdate(re)$，表示委托角色后主属性的更新操作函数。

（8）$onUpdate(Oa)$，客属性的使用中属性更新函数。

（9）$revoke(s,o,r)\Leftarrow\neg onE\vee\neg grant(r,s)\vee\neg contain(r,p)$。在执行过程中，一旦主、客体及主体所求权限不再满足委托函数，或者主体不再拥有包含特定权限的角色，那么主体获得的对于客体的访问权限将被系统收回。

3.3 模型的应用

RAUCON 包含以往传统的访问控制模型，如 MAC、RBAC、UCON。接下来，以书籍的在线阅读和智能卡访问控制管理说明 $RAUCONO_{123}$模型的实用性。

3.3.1 在线阅读实例

线上带委托 DRM 策略，用户拥有以下的操作权限：阅读的同时具有委托以及接受委托的权利。比如，两个用户 User1 和 User2，而在线书籍分为上部和下部两部分，阅读两部分需要分开付费。User1 同时购买了上部和下部两个部分，而 User2 只购买了读物的上半部分。两个用户同时在线阅读，当 User1 读完上半部分之后，由于市区对下半部部的阅读兴趣而将下半部分的阅读权限委托给了其他用户，也就是 User2，此类操作对应于 $RAUCON_{0123}$ 核心子模型体系中的 $RAUCON_{on32}$。具体的逻辑过程描述如下：

（1）用户 User1 与用户 User2 分别对应委托人与委托对象，客体资源 O 即为下半部网络书籍，P 则代表对下半部分数据的操作权限。

（2）User1 想要将对下半部分书籍操作的权限授予 User2，首先向授权系统 $RAUCON_{0123}$ 提交委托请求：$request\{User1, O, P, User2\}$。

（3）$RAUCONO_{123}$ 首先判断用户 User1 的请求是否合法：

$check\{User1, e, O, P\} \rightarrow \{true, false\}$，若返回 false，则拒绝请求。反之继续往下执行。

（4）明确委托属性：$e = (Oa, P)$。

（5）委托请求成功后，对此次涉及委托操作的主体属性进行更新操作：

$onUpdate(User1, Sa, Se)$， $onUpdate(User2, Sa, Se)$。

（6）进行委托操作：

$allowed(s, o, p) \rightarrow onD \wedge grant(r, s) \wedge contain(p, r)$。

3.3.2 智能卡访问管理实例

在智能卡某次会话中，有如下控制方式可选择：

（1）即使在同一次会话中，每次进行数据访问之前，都要进行主体身份及相关属性的验证操作，及 $RAUCON_2$ 模型。每一次的访问操作都进行安全环境的检测，可以确保非法用户进入系统。

（2）依据 $RAUCON_{0123}$ 建立智能卡访问的部分控制策略，如表 3.2 所示：

表 3.2　访问控制策略

智能卡	$RAUCON_{0123}$	采用模型
个人信息	A:Read,Write,Update B:写日志 C；插入卡片，建立通信	RAUCONon0、RAUCONpre1 RAUCONon2/RAUCONpre2
证书信息	A:Read B:写日志 C；插入卡片，建立通信	RAUCONon0、RAUCONpre1 RAUCONon2/RAUCONpre2
签名私密	A:Write B:写日志 C；插入卡片，建立通信	RAUCONpre0、RAUCONpre1 RAUCONon2/RAUCONpre2

表中第二列中的 A 代表授权,B 代表义务,C 代表条件。而第三列中的模型均 $RAUCO_{N0123}$ 核心模型体系中的部分子模型。

加入某用户 User 请求访问个人信息。首先系统需要检测主体所求权限是否符合条件约束：$preConChecked : preCL \rightarrow \{true, false\}$，例如是否写完日志、是否进行了双向认证等。如果返回 true，则进行下一步，否则系统将拒绝该用户的访问请求。

（3）主体请求访问满足系统授权规则的要求，系统授予主体访问个人信息的权限：

$$allowed(s,o,p) \rightarrow preA(s,o,p) \wedge preB(s,o,p) \wedge preC(s,o,p) \wedge grant(r,s) \wedge contain(p,r)$$

3.4 本章小结

本章针对传统 UCON 在委托方面表现出的不足，通过将 RBAC 的部分概念进入其中，提出一种具有委托性质的使用控制模型，初步解决了委托问题。在 RAUCON 模型的授权过程中，当主体对客体提出相应的访问操作请求时，系统会根据预、执行中职责，预、执行中条件，预、使用中委托三个方面因素授权决策。本章比较详细的对 RAUCON 的核心子

模型进行逻辑表述，并结合结合两个一个具体应用实例说明此模型的实用性。

第四章 具有时间约束的 RAUCON 模型研究

本章将在上一章提出的至于属性 RAUCON 使用控制模型的基础上，进行扩展，添加时间约束概念，关联起主属性和时间约束，扩展 RAUCON 模型的授权约束。在设计中考虑到授权实现会受主体属性变化的影响，从而进一步提升该模型的安全性和表达能力。

4.1 时限特性分类

在对 RAUCON 模型进行扩展之前，先简单描述几类典型时间约束。具体来说主要有激活时间范围内、长度约束和时间范围内激活时间长度约束。

（1）激活时间范围限制

此约束规定了用户、角色或权利（生效）激活的特定时间段。例如某公司的数据访问系统中，只允许职员在特定的时间段登录系统进行操作，而其他时间不允许某些用户登录系统。

（2）激活时间长度限制

此约束规定了用户、角色或权利生效（激活）的固定时间长度。此约束大多用在较敏感信息的系统，为防止用户登录时间过长而造成不必要的信息泄露。

（3）时间范围内时间约束长度限制

此模型规定了在一个固定的时间段内用户、角色或权利生效（激活）的累积次数。例如，某个信息数据管理系统规定，下午 3 点到 5 点之间，相同用户的登陆次数需在三次以内，总登陆时长在 40 分之内，此过程设立用户登陆计数器来实现，当计时器记录的登陆次数、总时长超过规定时限时，用户将被禁止登录系统。

另外，时间约束还有会话时限约束（可以影响一个或多个会话），周期时间约束[31]。

在模型中引入时限之后，一方面可以控制主体跨域访问的次数以及时间长度，另一方面可以将授权约束与时间进行更好的关联，提高模型的安全性和表达能力

4.2 具有时间约束的 RAUCON 模型

4.2.1 基本决策模型

在传统决策模型包括策略执行点（PEP）和决策点（PDP）两个基点以及策略库。在策略执行点用户提交请求，然后该请求被执行点提交给策略点，最后策略点根据策略库中的授权策略对用户相关授权。

以上是传统控制决策模型，如图 4.1 所示：其系统结构较为简单，且授权决策依据比较单一，仅仅依据部分主体属性就能够做出授权决策。此模型为决策基础模型，为访问控制提供了模型框架。

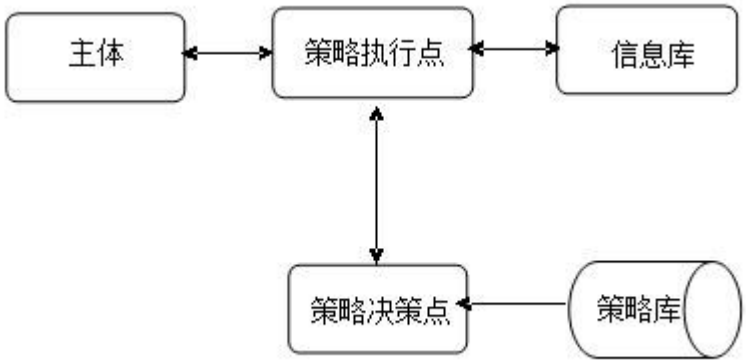

图 4.1 基础控制决策模型

访问控制过程中基础决策模型逻辑步骤描述如下：

（1）策略执行点接收到用户权限请求，包括要访问的客体以及相关操作等信息；

（2）执行点依据接受到的用户请求信息在信息库中取得相关属性，包括主、客体资源、环境属性等；

（3）执行点用户请求及上步取得的相关属性发送到策略决策点；

（4）决策点依据接受的信息从策略库中取得对应的策略；

（5）决策点将上步取得的策略返回给执行点；

（6）策略执行点依据返回的结果判决是否授权。

4.2.2 RBAC 模型中的时间约束

将时间引入授权决策之中，使系统在对用户进行角色指派时会将带有时限特性的角色指派给用户。包含时间因素的授权规则会依据主、客体属性（可信度）以及系统属性规定主体对客体的访问时限。我们用一个二元组（t_i, t_j）表示时间区间 T。t_i 与 t_j 分别表示时间段 T 的开始和结束。访问控制中时限的逻辑含义是只是在时间段 T 内才将包含访问和特定权限的角色 R 授予主体 S，一旦访问时间超出时间区间 T 的范围，系统将撤销主体 S 拥有的角色。

本章将依据权限授予和回收分离的思想，将 RBAC 中的时限特性部分进入 RAUCON 模型，对 RAUCON 模型进行扩展，使其在一定程度满足跨域系统访问控制的要求。

具体思想是将时限等级与其他属性进行关联，比如可信度、上下文等属性。由于跨域访问中目标域的时限具有可变性，可以依据主体的可信度等属性与目标域中的系统条件进行判定，主体每一次成功跨域访问都会被记录下来，然后目标域依据主体的访问记录对主体的可信度等重要属性进行相应的修改，以便对主体下一的跨域访问进行有效的时间控制。同时将主体的信任值提交给源域，作为一条重要的参考值方便源域对主体域内访问请求做授权操作。逻辑定义描述如下：

（1）$TSA(\text{Timed Subject Assignment}) \subseteq S \times R \times T$，表示主体的时限性指派即系统为主体指派包含激活时键约束的角色，且其中的时间约束具有可变性，由用户依据主体、客体资源以及系统条件等因素来确定。

（2）$T \subseteq t \times t, (t_i, t_j) \in T$，其中 T 为权限激活的时间区间，只有在此区间间内，主体被授予的角色才是有效的。

4.2.3 具有时间约束的 RAUCON 访问控制决策模型

本节将在前两节的基础上，在时间约束和角色两个方面对访问控制基础模型进行扩展，具体结构如图 4.2 所示。

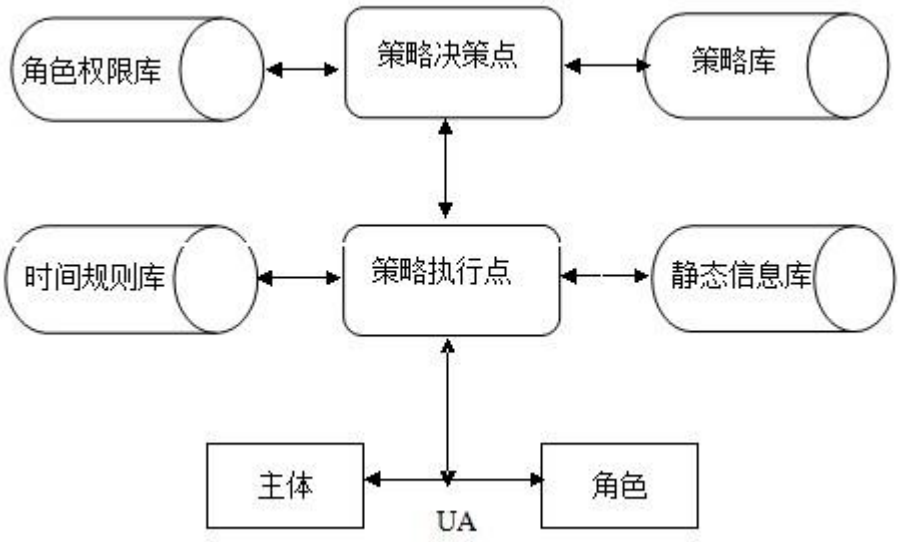

图 4.2　基于时间约束的访问控制决策模型

时间规则库中主要定义了时间约束与主、客体属性及系统条件间的对应关系。例如时间约束与主体的权限级别之间的关系，当主体的级别较高，系统就会提供较优的访问时间段或时长，反之则会对主体的访问时间段以及时长进行限制等等。该决策模型的控制逻辑步骤如下：

策略执行点接收到用户权限请求，包括要访问的客体以及相关操作等信息；

（1）执行点依据接受到的用户请求信息在信息库中取得相关属性，包括主、客体资源、环境属性等；

（2）执行点用户请求及上步取得的相关属性发送到策略决策点；

（3）决策点依据接受的信息从策略库中取得对应的策略；

（4）决策点将上步取得的策略返回给执行点；

（5）策略执行点依据返回的结果判决是否授权。

以上的扩展仅考虑到域内访问的情况，考虑到当今跨域访问的趋势性，现简单给出跨域访问下的控制决策模型。这需要在 RAUCON 模型的基础上进行跨域访问控制的扩展，在下一步工作中进行，在这里只给出跨域访问的决策模型。具体结构如图 4.3 所示。

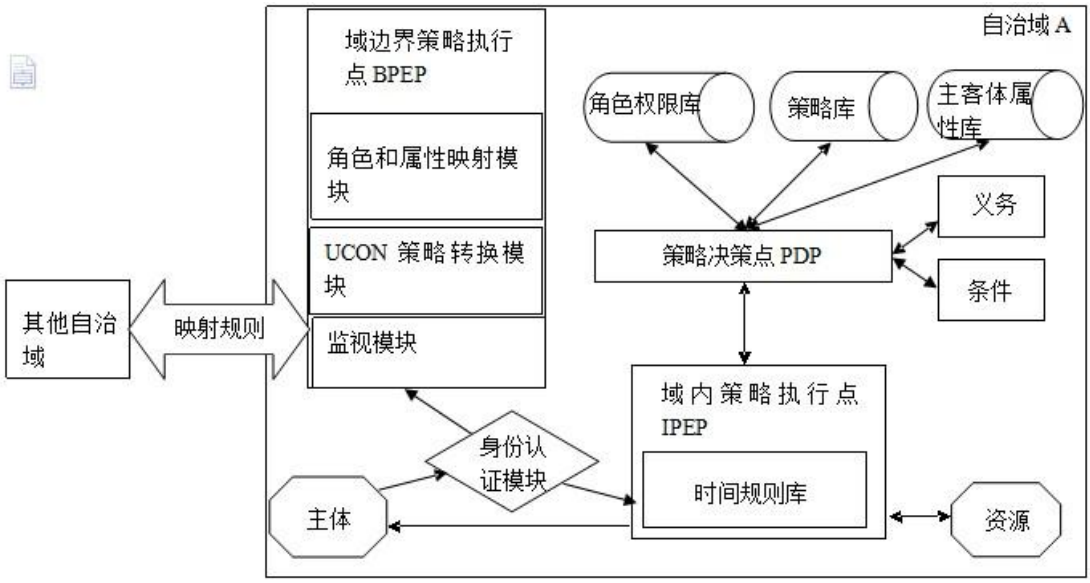

图 4.3 基于时限的跨域控制模型

其具体逻辑步骤如下：

（1）源域中的主体 S 准备队目标域中的某资源进行访问，首先将本域的边界策略执行点提交访问请求，其中包含主、客体属性，具体操作等主要信息，在这以前还有一步通过源域身份验证模块对主体身份的验证操作，验证的依据是系统中预设的 PKI 信任关系[34]，包括信任链表或信任列表；

（2）边界决策执行点将访问请求提交给本域的策略决策点；

（3）在这一步中，策略决策点将进行一系列的判决，包括主体属性是否满足相关访问条件，主体 S 是否履行完系统要求的义务，判决的依据是从策略库中取得与该访问请求有关的策略规则。若主体满足策略规则，系统将授予主体 S 对客体资源的访问权限并得出包含该权限的角色集合，将最后的授权决策结果提交给本域的边界决策执行点。反之，系统将拒绝主体对客体资源的访问请求，并向主体 S 返回授权判决结果；

（4）源域与目标域建立安全连接，源域边界决策执行点将用户证书（其中包含了主体 S 的属性以及可指派的角色集）和属性证书提交给目标域的边界决策执行点；

（5）目标域的边界决策执行点对角色或属性进行映射，取得源域中主体在目标域中的角色集或属性集。然后将取得的角色集或属性集生成决策请求，

向本域策略决策点提交；

（6）目标域的域内策略执行点再次对访问请求做条件与义务的决策判定，若满足相应的授权规则（即同意授予主体 S 访问的权限或相应角色），则同时向边界决策执行点和域内决策执行点发送判决结果和主体及当前环境属性。域内决策执行点将执行对主体 S 的授

权操作，边界决策执行点负责监控主体 S 的整个访问过程。反之将拒绝主体 S 的访问请求，并将判决结果提交于边界策略执行点。

模型通过对主体 S 的属性及访问请求等因素来制定时间约束。系统依据主体 S 的在跨域访问中的操作进行记录并评级，一次多为该主体下次访问请求的授权参考。

授权的时限性要求模型必须有相应的授权撤销机制。系统通过在执行过程中对主体做授权判决，当主体 S 访问时长已不再满足特定的授权规则，主体 S 的包含相应访问权限的角色就会被回收，这是一种自触发过程。

4.3 安全性分析

RAUCON 模型在引入时间约束之后授权管理的安全性，特别是域间授权管理得到了很大的提高。主要体现在：1.对主体的访问时间进行限制后，杜绝了因为无序访问导致的信息泄露等安全隐患。2.通过监控主体的活动，提高授权管理的反馈能力，一旦主体有越权等非法操作时，可以通过修改角色或属性映射，及时收回相应主体的操作权限。3.因为包含目标域内操作权限的角色集为映射所得，次角色的激活需要相关条件和属性满足特定的授权规则，提高了各域对域内敏感资源的保护能力。

4.4 本章小结

本章首先分析了时间约束在控制模型中的应用，将时限特性进入 RAUCON 模型中，提出了基于角色的时限性使用决策模型，并初步论述 RAUCON 模型进入时间约束后在跨域访问控制中的逻辑步骤，为接下来的研究工作奠定基础。

第五章　基于属性 RBAC 的使用控制模型 RUCON 的应用

5.1 系统概述

百名专家教授联百企”活动。活动旨在为党外专家教授提供实践锻炼平台，帮助非公有制企业解决生产经营中的技术难题，促进科研成果向生产力转化，打造统一战线服务科学发展新品牌。健全完善以企业为主体的产学研合作模式，建立以市场导向为主的技术转移和成果转化机制，及时把更多的科技创新成果转化为现实生产力，为企业加快发展和推进全省经济“转调创”提供强大动力。各高校教授以及附属研究院所，利用自身专业技术优势，帮助企业改进工艺，提高产品质量，降低开发和运营成本。高校的专家也可通过本章在上述研究的基础上，讨论 RUCON 模型在校联企业服务管理系统中的应用与实现。

5.2 系统设计

本文校联企服务管理系统涉及的单位有企业 A，学校 C 和 D。涉及的部门有、科研部、生产部、企业决策部、产品销售部、质检部等。针对某一企业的技术难题解决或产学转化，涉及的到多个院校的合作，一个院校可能与多家企业联系，一家企业也可能与多所院校关联，所以校联企服务管理系统是一个复杂的多域系统。本文提出的 RUCON 模型可以解决校联企服务管理系统中跨域协同的访问控制。

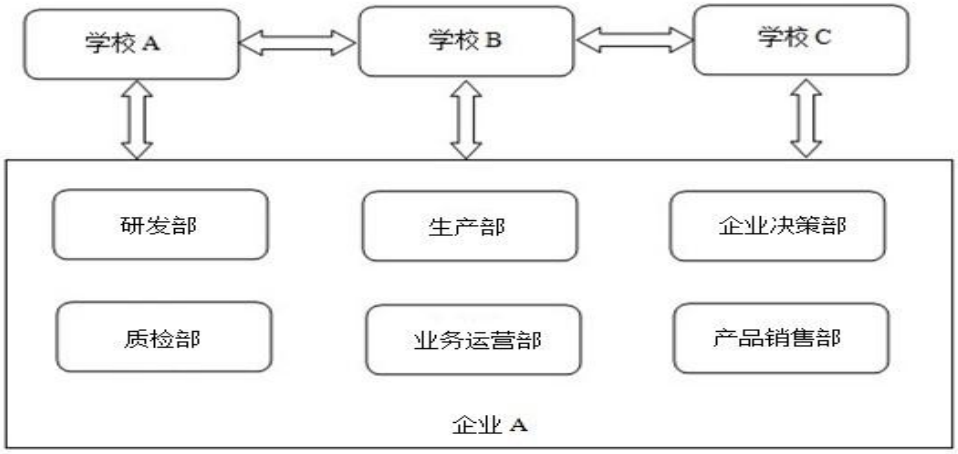

图 5.1　校联企服务管理系统跨域环境

校联企服务管理系统划分为系统管理、研发管理、运营管理、科研协作和技术交流、项目管理、科研成果管理等等。系统采用 RUCON 控制模型实现校联企服务系统的跨域访问控制。校联企服务管理系统的访问控制包括管理模块（用来实现属性、角色管理以及映射）、跨域控制处理模块（实现域间安全连接、接受用户认证信息、完成属性及角色映射）、请求判定模块（对用户的访问请求进行条件、责任等使用策略判断，如果授权决策条件满足，则对用户授予权限）、控制信息库（属性映射表、角色映射表、责任、义务等决策信息表，用数据表来存储属性、角色、权限、时限之间的关系表示）。其模块结构如图 5.2 所示。

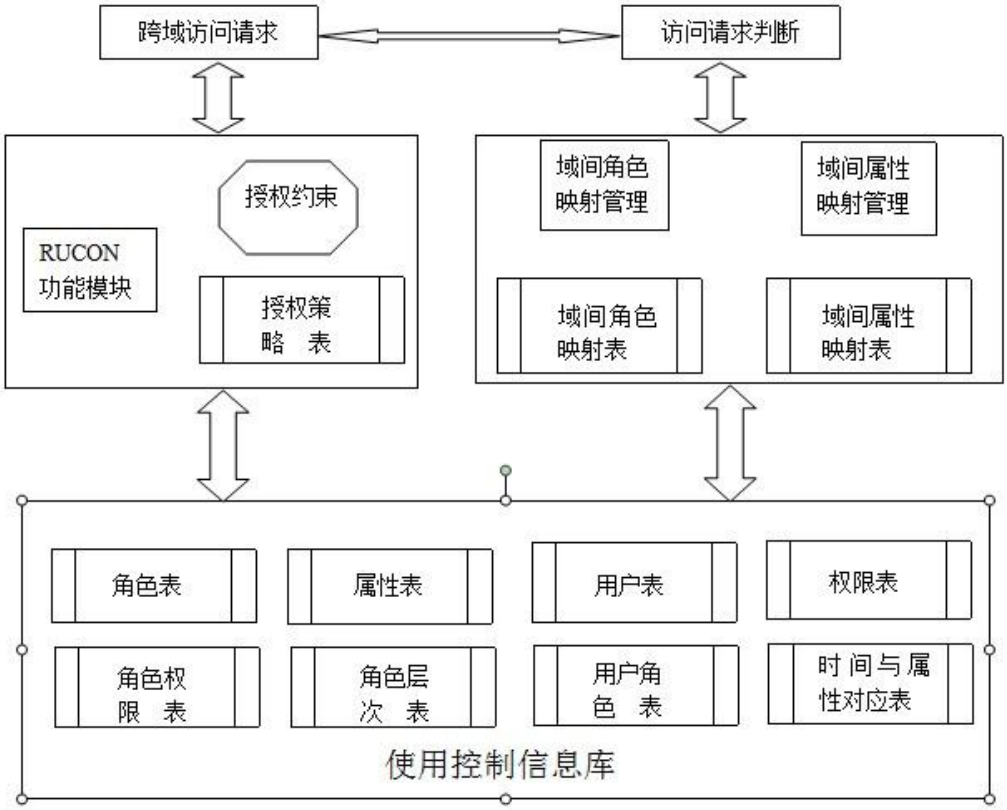

图 5.2　访问控制模型结构图

5.3 系统应用实例

学校 A 的用户某设计研究所 User1 访问企业 B 中的资源时，首先登陆学校 A 的校联企服务管理系统，此时系统会对 User1 进行身份验证，判断 User1 的 IP 是否在规定范围内、是否已阅读教育服务系统的使用说明，是否是在规定的时间段登陆，如果各授权决策条件满足，则允许 User1 进入系统并授予相应的访问权限，然后 User1 向校联企服务

管理系统的管理模块申请跨域访问企业 B。学校 A 与企业 B 建立安全连接，User1 的认证信息、角色信息等通过跨域控制模块传给企业 B 的跨域控制模块，认证通过后，企业 B 对 User1 进行属性或角色映射，请求判断模块对 User1 进行责任、义务、条件等使用决策元素判定，条件满足则赋予相应的访问权限。系统界面如图 5.4 所示：

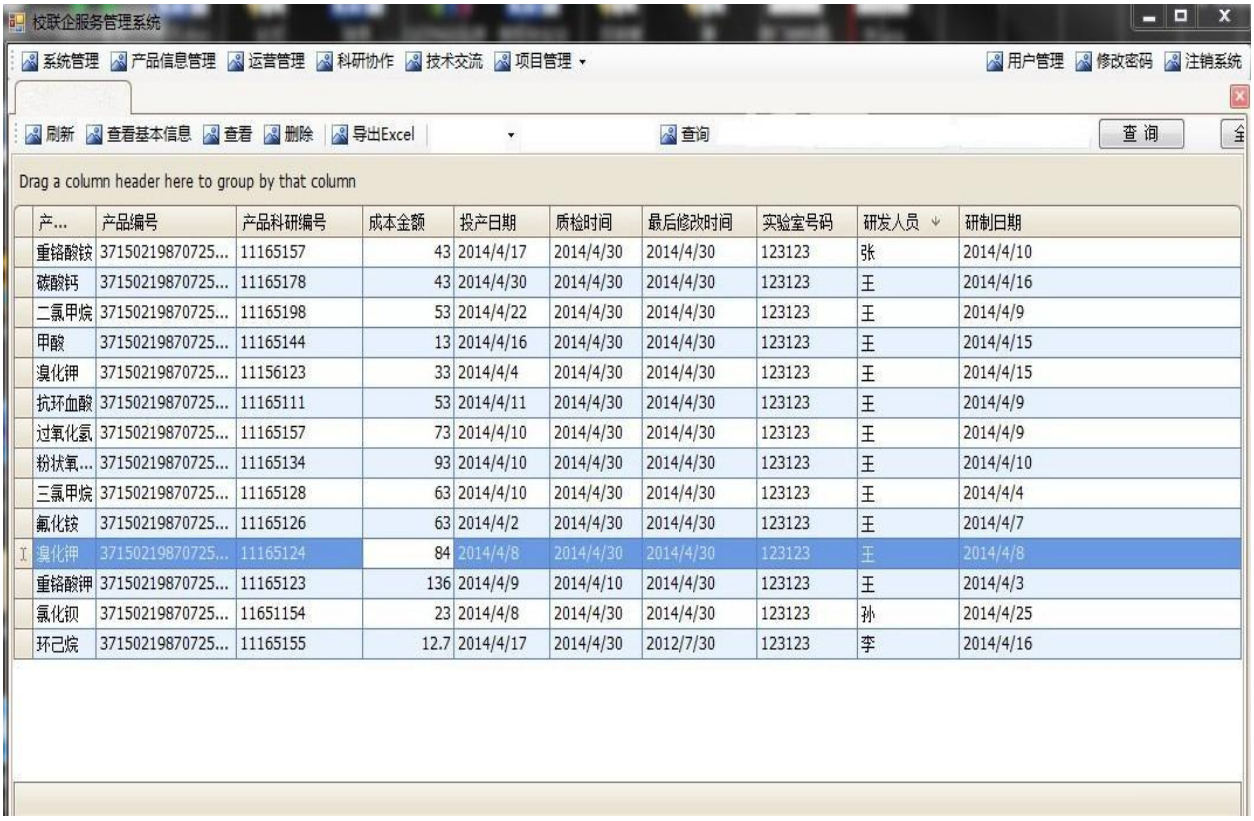

图 5.4　系统界面

学校 A 设计研究所与企业 B 的用户角色映射如图 5.4 所示。

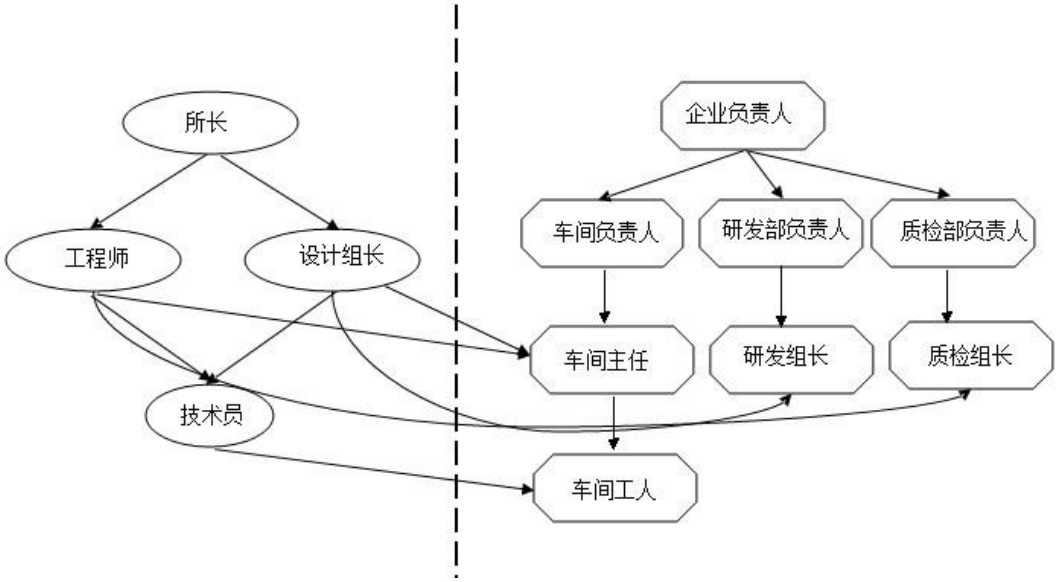

图 5.4　高校 A 与企业 B 的映射关系图

系统角色映射如图 5.6 所示

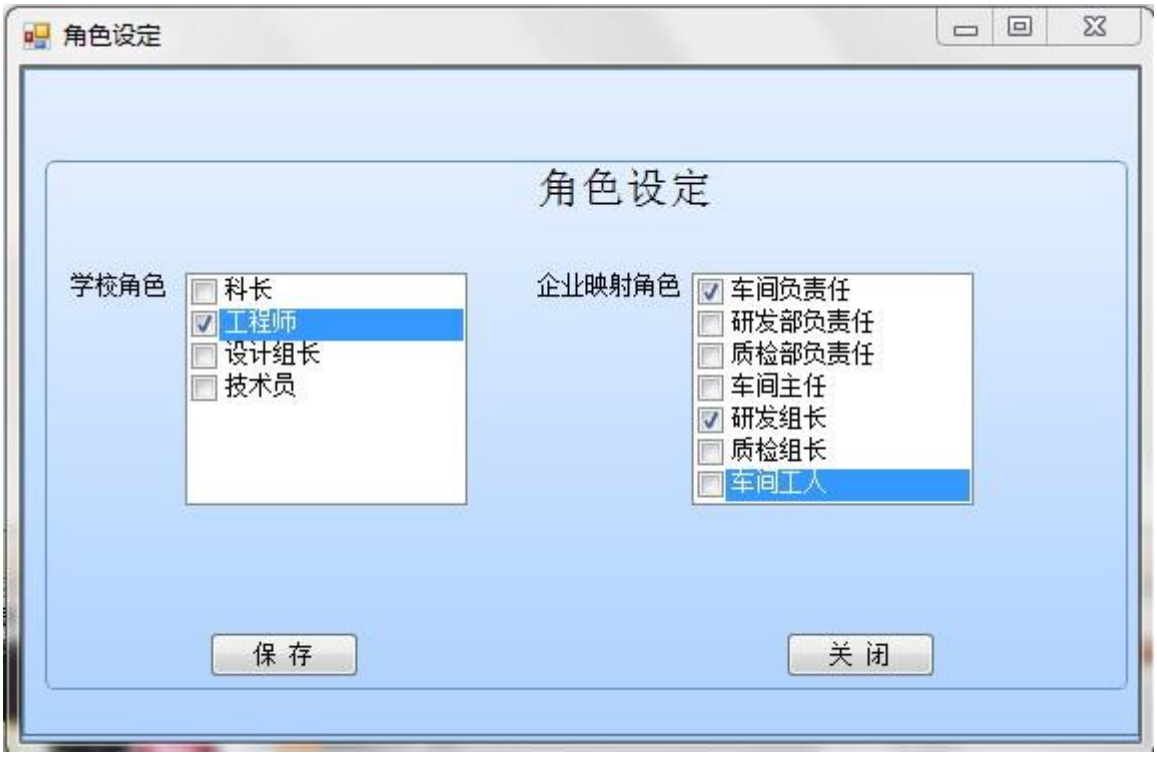

图 5.6　映射角色设定界面

若 User1 获得企业研发组长的角色，可以拥有（1）提出产品的设计思路与建议（2）查看产品信息，分析产品技术参数。（3）产品研发技术难题分析等权限。本系统会对用户的访问时间加以限制，并根据用户的信任度来制定主体跨域访问的时间。在目标域中，通过统计用户的历史访问记录来拟定主体的信任级别。信任度与时限的关系表 5.1 所示。

表 5.1　信任级别与时限对应表

信任度区间	信任级别	时限
0.8—1	完全	无时限
0.6—0.8	一般	不超 5 小时
0—0,6	不	不允许

5.3 本章小结

本章阐述了校联企服务管理系统的跨域访问环境，构建校联企服务管理系统结构，应用具有时限的 RUCON 模型。在实例中，体现了模型角色的分类、角色层次结构、跨域角色映射机制、具有时限的连续性授权决策，符合动态、细粒度的访问控制要求。

第六章 总结与展望

6.1 研究工作总结

计算机、网络以及流媒体等技术的发展以及将要来临的 4G 移动通信技术的大面积推广，数据信息以及服务的安全性越来越来成为影响技术应用的关键和人们关心的焦点。当今网络分布式、开放式的发展趋势，更加使得现有访问控制技术难以维持，需要更加智能，更加主动、更加灵活的控制模型。当今几类较为主流的访问控制模型尽管在一定程度上达到访问控制的要求，但随着技术的更新换代，终将会被取代。本文针对当下比较热门的使用控制模型 UCON 在角色委托及属性委托方面存在的不足之处，将 RBAC 模型中的角色概念进入到 UCON 中。给出 RAUCON 的结构关系，并形式化描述出 $RAUCON_{0123}$ 核心模型的各个子模型，包括授权模型 $RAUCON_0$、义务模型 $RAUCON_1$、条件模型 $RAUCON_2$、委托模型 $RAUCON_3$。通过将委托这一因素引入授权决策，使得在 RAUCON 中的主体能够通过将自己的属性或角色委托给其他的主体，实现 RAUCON 的委托功能。当今几种较为成熟的访问控制技术，提出一种新型的基于属性 RBAC 的具有委托功能的使用控制模型，主要工作有一下几个方面：

第一，提出一种给予属性的带委托使用控制模型 RAUCON，解决了 UCON 在委托方面的不足之处，该模型在吸收传统控制模型（如 RBAC 等模型）的优点，进入角色和委托等基本元素，完善 UCON 委托授权方面的不足之处。

第二，通过两个实例，验证了 RAUCON 模型在实际使用中的可行性。

第三，在给予 $RAUCON_{0123}$ 模型的基础上，进入时间约束要素，提出适合该模型适合的决策控制模型，并处给出了 $RAUCON_{0123}$ 核心模型在跨域访问控制方面的研究方向，进一步提高 RAUCON 模型的安全性，有助于该模型在现实生活中的使用，也为下一步对 RAUCON 模型的研究以及应用奠定基础。

6.2 待改进的工作

本文通过将 RBAC 中的角色和时间约束因素引入到使用控制模型 UCON 中，提出带委托功能和具有时间约束的使用控制模型 RAUCON。本文的研究取得了一定的成果，但仍然存在许多需要进一步研究的问题：

（1）没有对 RAUCON 模型进行系统的安全性分析，访问控制模型最核心、最根本的意义就在它的安全性，在应用到任何实际系统之前，所有访问控制模型都必须经过严密的安全性分析，所以对 RAUCON 的安全分析是接下来的一项主要工作。

（2）RAUCON 的管理模型尚待研究，RAUCON 作为一个使用控制的扩展模型，其相应管理模型的构建是保证模型实用性和完整性的关键。

（3）本文对于委托功能仅仅提出基础理论，对更进一步的角色委派关系和约束机制没有做深入的研究，这也使得 RAUCON 模型内容不够丰富，其适应能力不够强

（4）对模型中可能存在的并发性问题没有深入的探讨，因为在该模型的委托过程中，由于委托导致的属性更新可能具有多次性，即某一个属性的更新可能会导致其他属性同时出现更新操作，这个过程是并发进行的，如何确定属性更新的传递方向以及约束和管理整个更新的传递传递过程也将成为今后的研究重点。

（5）虽然在前面章节中，运用两个小实例验证了 RAUCON 模型在实际应用中的可行性，可这仅仅是在理论层面的探讨，RAUOCN 的研究很大程度上还是停留在理论阶段，抽象概念比较多。因此，根据当今的大型信息系统中的现实问题，给出较为具体的、能够真正满足现实需要的原型模型将是以后很长一段时间内需要去做的一项研究工作。

参考文献

[1]Masood A, Ghafoor A, Mathur A P. Conformance testing of temporal role-based access control systems[J]. Dependable and Secure Computing, IEEE Transactions on, 2010, 7(2): 144-158.

[2]Jarecki S, Saxena N. On the insecurity of proactive RSA in the URSA mobile ad hoc network access control protocol[J]. Information Forensics and Security, IEEE Transactions on, 2010, 5(4): 739-749.

[3] Waller A O, Jones G, Whitley T, et al. Securing the delivery of digital content over the Internet[J]. Electronics & Communication Engineering Journal, 2002, 14(5): 239-248.

[4]Valimaki M, Pitkanen O. Digital rights management on open and semi-open networks[C]//Internet Applications, 2001. WIAPP 2001. Proceedings. The Second IEEE Workshop on. IEEE, 2001: 154-155.

[5]Bertino E, Sandhu R. Database security-concepts, approaches, and challenges[J]. Dependable and Secure Computing, IEEE Transactions on, 2005, 2(1): 2-19.

[6]Yi-qun Z, Jian-hua L, Quan-hai Z. A general attribute based rbac model for web service[C]//Services Computing, 2007. SCC 2007. IEEE International Conference on. IEEE, 2007: 236-239.

[7]Park J, Sandhu R. Towards usage control models: beyond traditional access control[C]//Proceedings of the seventh ACM symposium on Access control models and technologies. ACM, 2002: 57-64.

[8]刘智敏, 顾韵华. 基于角色的跨域使用控制模型及其应用研究[J]. 信息技术, 2012 (4): 152-155.[9]Popescu B C, Crispo B, Tanenbaum A S, et al. A DRM security architecture for home networks[C]//Proceedings of the 4th ACM workshop on Digital rights management. ACM, 2004: 1-10.

[10]Kwok S H, Lui S M. A license management model for peer-to-peer music sharing[J]. International Journal of Information Technology & Decision Making, 2002, 1(03): 541-558.

[11]蔡伟鸿, 蔡建坤, 徐涛, 等. 基于属性 RBAC 及委托性质的使用控制模型[J]. 汕头大学学报: 自然科学版, 2010, 25(004): 57-65.

[12]杜光芹. 效用驱动的主题 Web 挖掘算法研究[D]. 山东师范大学, 2007.

[13]Sandhu R S, Coynek E J, Feinsteink H L, et al. Role-Based Access Control Models yz[J]. IEEE computer, 1996, 29(2): 38-47.

[14]Osborn S, Sandhu R, Munawer Q. Configuring role-based access control to enforce mandatory and discretionary access control policies[J]. ACM Transactions on Information and System Security (TISSEC), 2000, 3(2): 85-106.

[15]Sandhu R, Munawer Q. How to do discretionary access control using roles[C]//Proceedings of the third ACM workshop on Role-based access control. ACM, 1998: 47-54.

[16]Sandhu R, Park J. Usage control: A vision for next generation access control[M]//Computer Network Security. Springer Berlin Heidelberg, 2003: 17-31.

[17]Park J, Sandhu R. Originator control in usage control[C]//Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop on. IEEE, 2002: 60-66.

[17]Sandhu R, Park J. Usage control: A vision for next generation access control[M]//Computer Network

Security. Springer Berlin Heidelberg, 2003: 17-31.

[18]Park J, Sandhu R. The UCON ABC usage control model[J]. ACM Transactions on Information and System Security (TISSEC), 2004, 7(1): 128-174.

[19]Zhang X, Park J, Parisi-Presicce F, et al. A logical specification for usage control[C]//Proceedings of the ninth ACM symposium on Access control models and technologies. ACM, 2004: 1-10.

[20]Zhang X, Parisi-Presicce F, Sandhu R, et al. Formal model and policy specification of usage control[J]. ACM Transactions on Information and System Security (TISSEC), 2005, 8(4): 351-387.

[21]Rezgui J, Hafid A, Gendreau M. Distributed admission control in wireless mesh networks: models, algorithms, and evaluation[J]. Vehicular Technology, IEEE Transactions on, 2010, 59(3): 1459-1473.

[22]张娜. 基于机器学习的主题 Web 挖掘技术[D]. 山东师范大学, 2007.

[23]姚冬梅. 基于 UCON 的云计算访问控制模型研究[D]. 南京大学, 2012.

[24]Hebig R N, Meinel C, Menzel M, et al. A web service architecture for decentralised identity-and attribute-based access control[C]//Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, 2009: 551-558.

[25]Hur J, Noh D K. Attribute-based access control with efficient revocation in data outsourcing systems[J]. Parallel and Distributed Systems, IEEE Transactions on, 2011, 22(7): 1214-1221.

[26] Park S M, Chung S M. Privacy-Preserving Attribute Distribution Mechanism for Access Control in a Grid[C]//Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on. IEEE, 2009: 308-313.

[27]Kuhn D R, Coyne E J, Weil T R. Adding attributes to role-based access control[J]. IEEE Computer, 2010, 43(6): 79-81.

[28]Yuan E, Tong J. Attributed based access control (ABAC) for web services[C]//Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE, 2005.

[29]Lang B, Zhao N, Ge K, et al. An XACML policy generating method based on policy view[C]//Pervasive Computing and Applications, 2008. ICPCA 2008. Third International Conference on. IEEE, 2008, 1: 295-301.

[30]Yuan E, Tong J. Attributed based access control (ABAC) for web services[C]//Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE, 2005.

[31]Bobba R, Fatemieh O, Khan F, et al. Using attribute-based access control to enable attribute-based messaging[C]//Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual. IEEE, 2006: 403-413.

[32] Shamir A. Identity-based cryptosystems and signature schemes[C]//Advances in cryptology. Springer Berlin Heidelberg, 1985: 47-53.

[33]Boneh D, Franklin M. Identity-based encryption from the Weil pairing[C]//Advances in Cryptology—CRYPTO 2001. Springer Berlin Heidelberg, 2001: 213-229.

[34]Waters B. Efficient identity-based encryption without random oracles[M]//Advances in Cryptology–EUROCRYPT 2005. Springer Berlin Heidelberg, 2005: 114-127.

[35]Sahai A, Waters B. Fuzzy identity-based encryption[M]//Advances in Cryptology–EUROCRYPT 2005.

Springer Berlin Heidelberg, 2005: 457-473.

[36]颜学雄, 王清贤, 马恒太. Web 服务访问控制模型研究[J]. 计算机科学, 2008, 35(5): 38-41.

[37]Han J, Susilo W, Mu Y, et al. Privacy-preserving decentralized key-policy attribute-based encryption[J]. Parallel and Distributed Systems, IEEE Transactions on, 2012, 23(11): 2150-2162.

[38]Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007: 321-334.

[39] Kapadia A, Tsang P P, Smith S W. Attribute-Based Publishing with Hidden Credentials and Hidden Policies[C]//NDSS. 2007.

[40]陈颖, 杨寿保, 郭磊涛, 等. 网格环境下的一种动态跨域访问控制策略[J]. 计算机研究与发展, 2006, 43(11): 1863-1869.

[41]邓勇, 张琳, 王汝传, 等. 网格计算中基于信任度的动态角色访问控制的研究[J]. 计算机科学, 2010 (1): 51-54.

[42]夏启寿, 范训礼, 殷晓玲. 基于时间的 RBAC 转授权模型[J]. 西北大学学报: 自然科学版, 2009, 38(6): 932-936.

[43]道炜, 汤庸, 冀高峰, 等. 基于时限的角色访问控制委托模型[J]. 计算机科学, 2008, 35(3): 277-279.

[43] Misra S, Vaish A. Reputation-based role assignment for role-based access control in wireless sensor networks[J]. Computer Communications, 2011, 34(3): 281-294.

[44]许峰, 赖海光, 黄皓, 等. 面向服务的角色访问控制技术研究[J]. 计算机学报, 2005, 28(4): 686-693.

[28]初晓博, 秦宇. 一种基于可信计算的分布式使用控制系统[J]. 计算机学报, 2010, 1: 93-102.

[45] Krawczyk H, Lubomski P. Generalized access control in hierarchical computer network[C]//Information Technology (ICIT), 2010 2nd International Conference on. IEEE, 2010: 121-124.

[46] Ahmed T, Tripathi A R. Security policies in distributed CSCW and workflow systems[J]. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2010, 40(6): 1220-1231.

[47]朱圣刚, 刘欣, 韩臻. 时态数据库多级安全模型研究[J]. 计算机工程与应用, 2006, 42(20): 143-146.

[48]袁磊. 使用控制模型的研究[J]. 计算机工程, 2005, 31(12): 146-148.

[49]黄建, 卿斯汉, 温红子. 带时间特性的角色访问控制[J]. 软件学报, 2003, 14(11): 1944-1954.

[50]蔡伟鸿, 蔡建坤, 徐涛, 等. 基于属性 RBAC 及委托性质的使用控制模型[J]. 汕头大学学报: 自然科学版, 2010, 25(004): 57-65.

攻读硕士期间发表的论文

[1] 李阳,宋承祥,贾猛. 使用访问控制模型研究.山东师范大学学[J].山东师范大学学报（自然科学版）2013,9(2):10-13.

[2] 贾猛,宋承祥,李阳, 基于贝叶斯信任模型的资源调度策略[J]. 山东师范大学学报（自然科学版）,2013,22(3):685-688.

致 谢

本文完成之际，在此，首先我深深地感谢我的导师宋承祥老师。读研期间能跟随宋承祥老师学习，是我人生中最幸运的事情。导师渊博的学识，严谨的治学态度深深的感染着我，令我对访问控制模型研究产生了浓厚的兴趣。在读研究生的三年中，我从导师身上学到了扎实的专业知识和基本的研究方法。感谢导师带领我进入访问控制模型研究这样一个充满神秘和挑战的领域，一步一步的引导着我读论文，写论文，引导我步入科研的大门，尤其在论文的选题，写作及定稿过程中导师给予我悉心的指导和热情的帮助。宋承祥老师教会我怎样去寻找问题，怎样去解决问题，这将使我受益终生。

宋承祥老师是我人生的一大转折点，有了*老师的谆谆教导，我的人生将变得更加绚丽多彩。

衷心感谢山东师范大学信息科学与工程学院各位领导和老师三年来对我的关怀和帮助。

衷心感谢贾猛师兄在读研期间给我莫大的帮助，感谢他们在学习上给予的大力帮助。

衷心感谢我的同窗孟帅，他们在我研究生学期期间以及论文的写作方面给与我很大的帮助，帮助我解除疑惑。

衷心感谢我的师弟师妹在读研期间对我的帮助，虽然研究方向不一，但是他们所叙述的方法对我的论文有了很大的启发作用。

感谢我的父母，是他们在物质上和精神上给予了我极大的支持，使我能够顺利完成我的学业。

感谢所有关心、爱护、帮助和启发过我的朋友，感谢他们对我一如既往的支持和关爱。

衷心感谢大家！