第35卷 第14期 计 算 机 工 程 2009年7月
Vol.35 No.14 Computer Engineering July 2009
·软件技术与数据库· 文章编号：1000—3428(2009)14—0041—04 文献标识码：A 中图分类号：N945.1

基于扩展 PMI 的电子政务应用安全平台

高国伟，王延章
(大连理工大学管理学院信息与决策技术研究所，大连 116024)

摘 要：针对电子政务应用的安全问题，提出适用于扩展授权管理基础设施的政府组织元模型，给出相应数学描述，采用该模型构建应用开发平台，将其用于实际项目建设。应用结果证明，该模型能适应电子政务应用多变的安全需要，实现授权管理，具有较高推广价值。
关键词：电子政务；政府组织元模型；授权管理基础设施；应用安全

E-government Application Security Platform Based on Extended PMI

GAO Guo-wei, WANG Yan-zhang
(Institute of Information and Decision Technology, School of Management, Dalian University of Technology, Dalian 116024)

【Abstract】Aiming at the security problem of E-government application, this paper proposes the government meta-organization model for extended Privilege Management Infrastructure(PMI) and the relevant mathematical description. The application development platform is constructed by using this model, and is applied to the practical project establishment. Application results show that this model can adapt to E-government applications security needs in changing environment, realize authorization management, and has high promotional value.
【Key words】E-government; government meta-organization model; Privilege Management Infrastructure(PMI); application security

1 概述

随着电子政务的深入发展和广泛应用，政务系统的复杂程度越来越高，因此，电子政务应用的安全问题成为电子政务建设急需解决的关键问题之一[1]。

传统授权管理基础设施(Privilege Management Infrastructure, PMI)技术通过数字证书机制管理用户的授权信息，并将授权管理功能从传统应用系统中分离出来，以独立服务的方式面向应用系统提供授权管理服务。在某些应用和应用的某些层面上，为应用系统的设计、开发和运行管理提供了很大便利[2]。但在电子政务应用中，安全责任与岗位职责相对应，一些职权无法从应用的操作中剥离出来，导致 PMI 在实际应用中受到限制。

为了解决上述问题，需要通过扩展 PMI 使安全技术向应用层渗透，从而将安全技术与业务应用有机结合。

2 政府组织元模型

组织元模型的研究主要集中在企业的组织模型上[3-4]，对政府组织模型的研究较少。而政府组织与企业组织存在很多差异[5]。

通过以下定义描述政府组织元模型：

定义 1 组织可以表示为 $ORG=\langle ORG,USR,POS,ROL,BO,INF,TSK,GOL,PLC,TIM\rangle$，其中，$ORG$ 也可以是组织单元集合；USR 为成员集合；POS 为岗位集合；ROL 为角色集合；BO 为业务对象集合；INF 为信息对象集合；TSK 为任务集合；GOL 为目标集合；PLC 为政策集合；TIM 为时间点集合。

定义 2 用户可以表示为 $USR=<ORG,TEAM,POS,PEOPLE,AGENT>$，其中，$TEAM$ 为组织中面向过程的动态组织结构和构成单元；POS 为用户拥有的岗位；$PEOPLE$ 为组织中的所有人；$AGENT$ 为智能主体。

定义 3 业务对象可以表示为 $BO=<TOB,OPT,\cdots>$，其中，TOB 代表业务对象属性集集合；OPT 代表组织中可操作算子集合。

定义 4 角色集可以表示为 $ROL=<RIGHT,DUTY,RELATION>$，其中，$RIGHT$ 表示权限；$DUTY$ 表示责任；$RELATION$ 表示关系。$RIGHT$ 是集合 $TOB\times 2^{OPT}$ 的子集，$RIGHT$ 中的每个元素都表示一种权限。若 $(tob,opt)\in RIGHT$，其中，$opt\in OPT$，则表示对属性 tob 可以有一种访问权限，允许对 tob 进行 opt 操作。在安全政务系统中，角色是能控制一定应用资源的用户集合。

定义 5 对于角色 rol，权限 $right_1$，$right_2\in RIGHT$，其中，$right_1=(tob,\ opt_1)$，$right_2=(tob,\ opt_2)$，如果 $opt_1\cap opt_2\neq\phi$，则 opt_1 与 opt_2 是重叠的。

定义 6 $\forall rol_1,rol_2,rol_3\in ROL$，如果具有角色 rol_1 的用户可以把角色 rol_3 赋予具有角色 rol_2 的用户，则称 rol_1 对 rol_2 具有在 rol_3 上的指派关系，记为 $rol_1\xrightarrow{rol_3}rol_2$。

在基于政府组织元模型构建的系统中，多个协同用户具有不同权力和责任，指派关系提供用户自己进行权限分配的手段。

定义 7 $\forall tob_1,tob_2\in TOB$，$\forall opt_1,opt_2\in OPT$，如果任何用户都不能同时具有在 tob_1 上的操作 opt_1 或在 tob_2 上的操作 opt_2 的权限，则称 tob_1 上的 opt_1 与 tob_2 上的 opt_2 是冲突操作，记为 $opt_1(tob_1)\,|\,opt_2(tob_2)$。对于权限 $right=(tob,\ opt)$，其中，$opt\in OPT$，如果 $\exists opt_1\in OPT$，$\exists opt_2\in OPT$，则不允许 opt_1

基金项目：国家自然科学基金资助项目“电子政务模型体系及政务流程再造研究”(70271045, 60074038)
作者简介：高国伟(1973－)，男，博士研究生，主研方向：管理信息系统，软件工程，复杂信息系统；王延章，博士、博士生导师
收稿日期：2009-01-16 **E-mail：**myemailpost@sina.com

与 opt_2 是 tob 上的冲突操作。

定义 8 $\forall right_1=(tob_1,opt_1),right_2=(tob_2,opt_2)\in RIGHT$，其中，$tob_1,tob_2\in TOB,opt_1,opt_2\in OPT$，如果 $\exists opt_1,opt_2\in OPT$，且 $opt_1(tob_1)\,|\,opt_2(tob_2)$，则称 $right_1$ 与 $right_2$ 是冲突权限，记为 $right_1|right_2$。 $\forall rol\in ROL$，如果 $\exists right_1\in rol,right_2\in rol$，则不允许 $right_1|right_2$。

定义 9 $\forall rol_1,rol_2\in ROL$，如果 $\exists right_1\in rol_1,right_2\in rol_2$，且 $right_1\,|\,right_2$，则称 rol_1 与 rol_2 是冲突角色，记为 $rol_1\,|\,rol_2$。角色冲突的概念提供了一种对角色间关系的描述，规定了用户不能同时具有的角色，可以实现在角色分配时进行控制的目的，从而防止权力滥用。

定义 10 $\forall usr_1,usr_2\in USR$，如果 $\exists rol_1\in usr_1,rol_2\in usr_2$，且 $rol_1\,|\,rol_2$，则称 usr_1 与 usr_2 是冲突用户，记作 $usr_1\,|\,usr_2$。

定义 11 一个角色可以同时继承多个角色，称为角色间的多重继承。若 rol_1 继承了 rol_2，记为 $rol_1\prec rol_2$，rol_2 为 rol_1 的父角色，rol_1 为 rol_2 的子角色。

定义 12 一般地，一个业务处理过程 P 从满足预条件集 Φ 的某个时间点开始，到满足结束条件 Θ 的某个时间点终止。过程可以视为一种递归形式，并最终归结到元活动 *action*(在某个抽象层次上不可再分的活动)，可以表示为

opt(tob)= opt(tob); opt(tob)
| opt(tob)^ opt(tob)
| condition? opt(tob): opt(tob)
| (condition? opt(tob))*
| AND (opt(tob), opt(tob),…)
| OR (opt(tob), opt(tob),…)
| XOR (opt(tob), opt(tob),…)

这些关系可以在安全模型视图上依据不同的安全需求影射为针对性的安全关系。

定义 13 角色结构关系 RSR=<ROL, $f$rsr>, RLT={EQU,$EXLD$, CTN,NED,$SPRT$,$OVLP$…}，RLT 是一个集合，设 $RSR=ROL\ X\ ROL$, $RLT\subseteq RSR$，它表示角色之间常见的 7 种关系(在本文中不包括继承关系)。组织遵循如下基本规则：

规则 1 任何冲突的角色都没有继承关系，即

$$(rol_1\,|\,rol_2)\Rightarrow\neg((rol_1\prec rol_2)\vee(rol_1\succ rol_2))$$

规则 2 不存在可以多重继承的互斥角色，即

$$rol_1\,|\,rol_2\Rightarrow\neg((rol_1\prec rol)\wedge(rol_2\prec rol))$$

规则 3 一个角色继承的互斥角色的某一方与另一方互斥：

$$(rol_1\,|\,rol_2)\wedge rol\prec rol_1\Rightarrow rol\,|\,rol_2$$

规则 4 角色间的指派关系可以在具有继承关系之间的角色间传递，即

$$\forall rol_1,rol_2,rol_3,rol_4\in ROL,(rol_1\xrightarrow{rol_2}rol_2)\wedge(rol_4\prec rol_3)\Rightarrow rol_1\xrightarrow{rol_4}rol_2$$

规则 5 若角色的权限在业务过程中存在某种关系，这些关系必然会反映在它们所属的角色中，即

$$\exists right_1,right_2\in RIGHT,right_1\in rol_1,right_2\in rol_2,right_1=F_1(right_2)\Rightarrow rol_1=F_2(rol_2)$$

基于以上论述，得到角色的形式化描述如下：

<Role> ::= DEF_ROLE <Role id>
[NAME <name>]
[<Description>]
[<Users list>]
[<Info list>]
<Relation list>
<Duty list>
<Right list>
END_ROLE
< Users > ::= DEF_USERS <User id>
[NAME <name>]
[<description>]
[<formal parameters>]
END_ USERS
<Relation> ::= DEF_RELATION <Relation id>
<Relation_Statement list>
END_RELATION
<Relation_Statement list> ::= <Relation_Statement> [<Relation_Statement list>]
<Relation_Statement> ::= <Relation_Type> <Role id list>
<Relation_Type> ::= EQU | EXLD | CTN | NED | SPRT | OVLP |…
<Duty> ::= DEF_DUTY <Duty id >
[NAME <name>]
[<formal parameters>]
[WHEN <Event | Message | Command>]
IF <Condition>
THEN <Procedure>
[POST_CONDITION <Condition>]
END_OBLIGATION
< Procedure > ::= <Composite_Action>
| <Composite_Action> <ActionCombinatorOp> <Composite_Action>
| < Procedure > < ProcedureCombinatorOp> < Procedure >
<ActionCombinatorOp> ::= <SquenceActionOp>| <RelActionOp> |<ParaellActionOp>|…
<ProcedureCombinatorOp>::=<SquenceProcedureOp>|<Rel Proce dureOp>
| <ParaellProcedureOp>|…
Elementary_Action ::= <Operator> <BusinessObject>
<Composite_Action>::=<Elementary_Action><ActionCombinator Op>
< Elementary_Action >
<Right>::=DEF_PERMISSION <Right id>
[NAME <name>]
[<formal parameters>]
[<Elementary_Action > list]
END_PERMISSION
< Info >::=DEF_RELATION < info id>
< info list>
END_RELATION
< info _Statement list> ::= < info _Statement> [<info _Statement list>]
< info _Statement>::= < info _Type> < info _Content >
< info _Type>::= ROLE | BUSINESSOBJECT | BACKGROUD | BUSINESSOBJCETSTREAM | OPERATOR|…

3 电子政务应用安全体系架构

按上述政府组织元模型的描述，在政务应用层面需要对角色、任务和业务对象进行安全控制，如图 1 所示。该结构满足如下应用安全原则和需求：

(1)对用户进行访问控制。

(2)支持动态改变角色权限。

(3)支持协同权限的说明和控制，系统中除了普通的数据访问操作(如读、写)外，还包括有关用户交互、协作的访问操作，并提供对协同权限的相应控制。

(4)提供方便的授权和取消机制、操作合法性检查机制。

(5)用户之间的授权关系具有多用户交互和协作等特性，各用户可以授予其他用户某些权限，支持对操作依赖关系的描述。

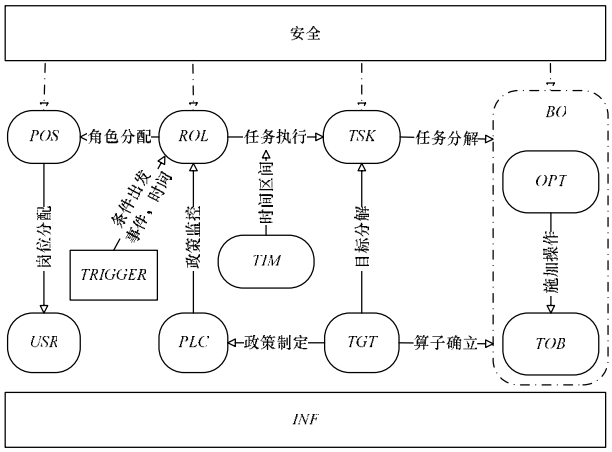

图 1　政府组织元模型中的安全控制

基于政府组织元模型的电子政务应用安全体系架构如图 2 所示。

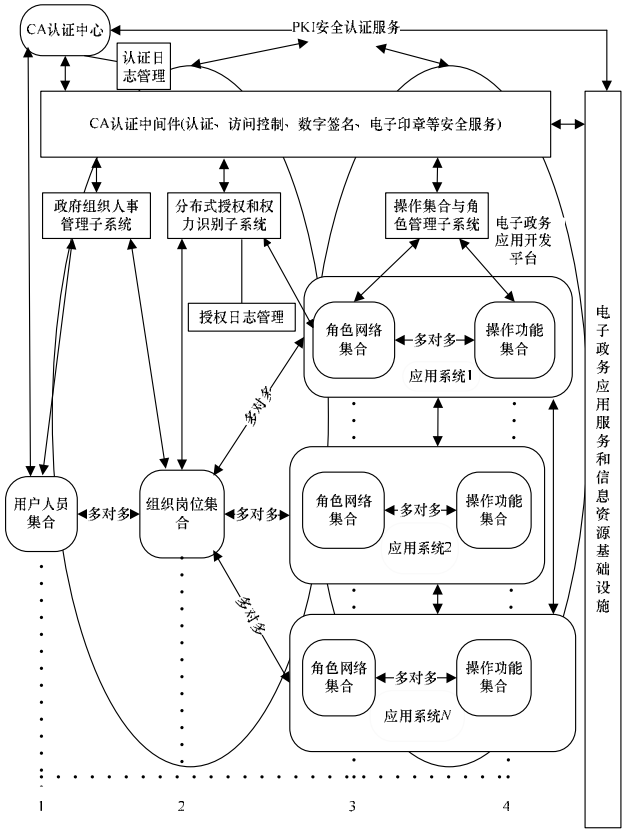

图 2　电子政务应用安全体系架构

第 1 层和第 2 层应用系统之间是独立于任何其他应用系统的组织人事管理系统。组织人事管理系统负责管理政府部门的所有人员、岗位信息和它们之间的关联关系，并将相关信息存储在组织人事数据库中。由于该系统不过多涉及技术问题，因此可以由行政部门管理，符合政府机构的行政专业化管理，且保证了应用安全。

第 3 层和第 4 层的应用系统管理各个系统的业务、操作集和角色信息，以及角色之间的对应关系。第 2 层和第 3 层之间是岗位和角色的对应关系，由分布授权和识别权力子系统完成。可以将政府组织人事合二为一，以方便系统的组织授权。

根据国家电子政务建设要求，将来所有电子政务信息系统都要和国家、省、市的 CA 认证中心进行集成。整个系统建设置于公开的密钥基础设施(Public Key Infrastructure, PKI)的安全体系之上。

4　电子政务应用安全集成平台结构与关键技术

4.1　电子政务应用安全集成平台结构

如图 3 所示，本文开发了基于组织元模型的电子政务应用安全集成平台(中间件)。其主要功能如下：用户通过岗位-角色拥有对应用系统中对象的权限。在权限管理中，可以将资源的权限直接指派到用户。电子政务应用系统中的所有用户都必须经过审核，发放数字证书。拥有证书的用户由 CA 系统生成，授权管理系统和 CA 系统共享一个 LDAP 服务器，其中存储了用户的一些基本信息以及与 CA 系统和授权管理有关的用户属性信息。授权管理中心维护整个系统范围的用户信息和属性，下级授权管理系统可以查阅并获得系统内所有用户的基本信息、证书信息。下级授权系统可授权的用户范围不能超过整个系统的用户范围。当下级部门有用户的添加和删除需求时，需要通知上级授权系统，由授权管理中心通知身份认证系统，身份认证系统负责用户的添加与删除。岗位-角色的委派方式由安全策略部门制定，委派过程由下级授权管理系统具体操作。用户可以属于一个岗位和一个或多个角色，能继承每个角色的权限，包括角色的添加、删除、角色属性的修改和岗位-角色的指派、岗位-角色权限的指派。

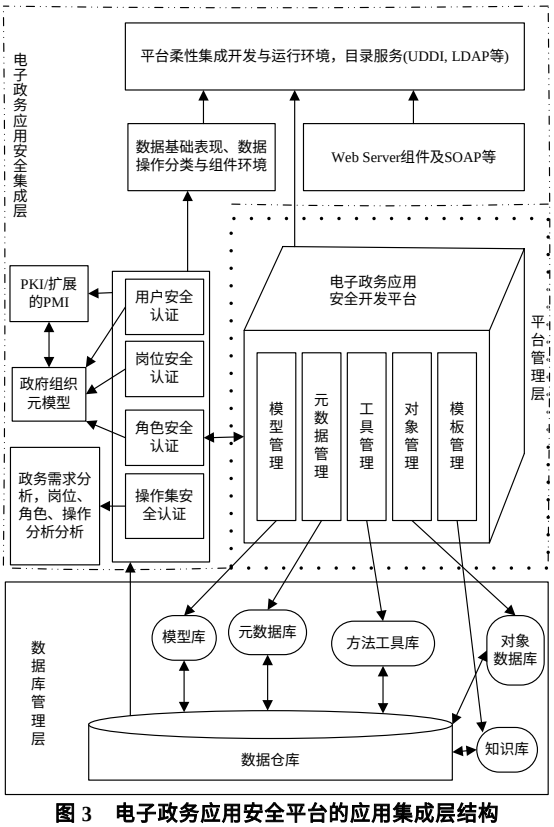

图 3　电子政务应用安全平台的应用集成层结构

4.2 关键技术

本文集成平台采用 J2EE 技术，开发相应的软件系统，涉及的开发平台工具如图 5 所示。其中，Struts Frameworks 是 J2EE 中较成功的开发框架；Struts 是一种较典型的 MVC 系统框架。电子政务应用安全集成平台基于 Struts 的 MVC 模式，对系统的各个组件进行组织。如图 3 所示，将业务逻辑、控制逻辑和显示逻辑分离，降低了系统各模块间复杂的耦合关系，提高了系统的灵活性、可维护性和安全性。电子政务应用安全的控制器是联系 Model 和 View 的纽带，它实现两者的松散耦合，并基于政府组织元模型实现角色之间的信息共享和协作的控制，如图 4 所示。

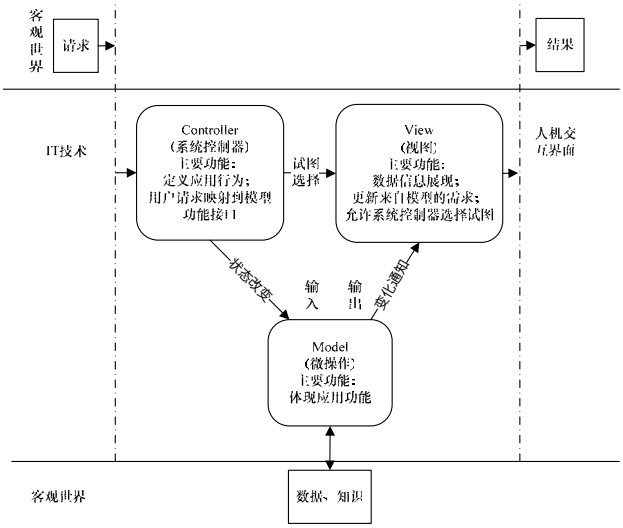

图 4　基于 Struts 的电子政务应用安全平台结构

组织元模型的主要功能如下：

(1)扮演各平行政府部门对应的下属角色网络，在相应的应用系统上进行通信和互操作，从而实现各平行部门应用系统与安全平台的集成。

(2)扮演单纯角色，实现某些业务的安全自动处理、推理和辅助决策等功能。

(3)实现登录认证、安全传输、加/解密、签名/验签等安全功能。

(4)记录用户的登录与注销、调用的系统功能与执行时间、访问的信息资源，并对其进行审计。

(5)为确保系统正常运行，对系统的运行状况，包括软件、硬件、网络等进行监控，并提供预警功能。

电子政务应用安全的控制器中的关键角色对象用 Java 描述如下：

```
Class Role
{String RoleName;//角色名称
 String type;//角色描述
 String Roledescription;//角色对应岗位
 String RoleOrg;//所属组织
 String superiorRole;//上级
 Vetor RoleAffairObjectList;//角色处理的业务对象及其属性
 Vetor RoleAffairObjectConstraint;//表示赋予角色关于处理业务
//对象的约束条件，即能对处理的业务应用的属性及相应权限，如插
//入、修改、删除、浏览、密级等。
 Vector RoleOperationList;//描述角色被赋予的操作
 Vector RoleOperationConstraintRule;//描述激发操作的关于业
//务对象的状态和判定规则
 …//角色的其他属性
```

电子政务开发平台的模型主要实现 2 个功能：(1)基于系统的建模工具实现对系统的建模。系统建模包括岗位角色管理构件、用户管理构件、功能目录服务管理等。(2)信息展现组件作为开发平台的试图 View，面向最终用户实现多样化的直观显现。

Hibernate Frameworks 是目前 Java 领域内功能性、稳定性较高的 I/O 映射处理框架。使用该框架可以充分使用面向对象的程序设计方法，通过 Java 类与关系型数据库建立关联，实现 Java 对象的自动存储、高效率的缓冲和完善的事物处理，保证了系统的稳定。图 5 描述了系统开发关键技术。

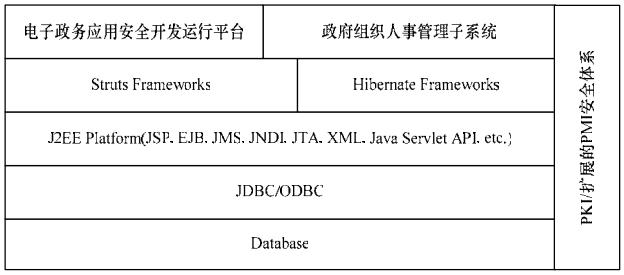

图 5　系统开发关键技术

5　应用案例

笔者采用上述结构成功完成了多个地区和城市的电子政务应用并取得了较好效果，主要包括：大连市市委、大连市政府、大连市中山区、西岗区政务的应用系统，杭州市政务内网办公资源管理系统，绍兴内网门户网站系统。

本文以大连市西岗政府为例进行说明，该系统的用户涉及区委、人大、区政府、政协、街道相关企业等 100 多个点。应用系统包括政务外网门户网站、领导辅助决策支持系统、内部办公系统、公文交换系统、内容管理系统、会议汇报系统、即时消息、邮件管理、日程安排、工作计划等。该系统结合大连的 CA 体系，利用组织人事管理系统实现了单点登录、统一认证、统一授权，并通过目录交换与服务技术实现信息系统的真正融合以及信息资源的安全共享。

参考文献

[1] 国家信息化领导小组. 国家电子政务总体框架[Z]. 2006.

[2] 国家信息安全工程技术研究中心. 电子政务总体设计与技术实现[M]. 北京: 电子工业出版社, 2003.

[3] TOVE. TOVE Manual[Z]. Enterprise Integration Laboratory, University of Toronto, 1995.

[4] Uschold M, Gruninger M. The Enterprise Ontology[J]. The Knowledge Engineering Review, 1998, 13(1): 31-35.

[5] Rosenbloom D H. Public Administation: Understanding Management, Politics, and Law in he Public Sector[M]. 5th ed. [S. l.]: McGraw-Hill Companies Inc.. 2002.

编辑　陈　晖