基于信任度的网格虚拟组织访问控制模型*)

崔永瑞[1] 李明楚[1] 胡红钢[2,3] 任一支[1]
(大连理工大学软件学院 大连 116621)[1] (中国科学院软件研究所 北京 100080)
(信息安全国家重点实验室 北京 100049)[3]

摘 要 针对现有网格虚拟组织访问控制模型欠缺描述上下文约束的能力、资源端管理负担沉重以及未能刻画成员间的真实信任关系等不足,给出一种基于信任度的网格虚拟组织访问控制模型 TwBAC(Trustworthiness-based Access Control model),该模型能够刻画带有上下文的访问控制策略;对资源实体进行抽象,减轻管理负担;应用信任度刻画虚拟组织成员之间的信任关系,并有效控制委托深度。此外,采用分布式管理模型 AdTwBAC 实现"虚拟组织成员域管理自治",并结合具体应用实例进行了说明。

关键词 TwBAC 模型,信任度,虚拟组织,访问控制

Trustworthiness-based Access Control Model for Grid Virtual Organizations

CUI Yong-rui[1] LI Ming-chu[1] HU Hong-gang[2,3] REN Yi-zhi[1]
(Software School, Dalian University of Technology, Dalian 116621, China)[1]
(Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)[2]
(State Key Laboratory of Information Security, Beijing 100049, China)[3]

Abstract The existing access control models for grid virtual organization(VO) have some faults as follows: First, most of them cannot express contextual access control policies which are important in the grid environment. Second, they bring heavy administration burden at the side of resource providers. Finally, trustworthiness between VO members is not described in these models which do not accord with facts. To overcome these limitations, provided a trustworthiness-based access control model for virtual organizations, called TwBAC model. It can express contextual access control policies, and offer the similar abstraction for resource objects as roles for subjects. Besides, expressed the trust relationship of VO members accurately by using trustworthiness, which can also be used to control the depth of delegation. Presented the model, administration model and an application case.

Keywords TwBAC model, Trustworthiness, Virtual organization, Access control

1 引言

虚拟组织(VO)是网格的核心概念之一,它由任务请求者发起,通过规范访问控制策略来刻画共享关系,规定资源使用方法与上下文,进而划清虚拟域的逻辑边界,将不同的成员组织中的合法用户和资源包含进来,实现大规模、跨组织资源共享与协作[1]。由于虚拟组织的用户与资源数量巨大,且包含了过多的访问控制宿主,使得访问控制复杂度大大提高。另外,每一个成员域都有权在不需外界知晓或干涉的情况下对本地成员进行访问控制,因此如何执行访问控制策略,对虚拟组织的生成与运行进行高效安全的管理,成为保证这一动态共享环境安全运行的关键。

一个安全高效的虚拟组织访问控制模型应具备如下特点:(1)能够刻画虚拟组织中的各类实体(用户、资源等),建立抽象逻辑层;(2)能够刻画带有上下文约束的访问控制策略,适应动态变化的虚拟组织环境;(3)能够结合信任度来准确刻画虚拟组织中各成员间的信任关系;(4)其管理模型应能规范成员域的管理责任,以及虚拟组织访问控制策略的生成与管理,灵活地对不同的管理职责进行建模;(5)能够将策略与动态的底层基础设施相分离,并将管理任务分派给成员域处理,实现各成员域的管理自治。本文给出的 TwBAC 模型较好地满足上述需求,以该模型为基础,可以动态生成并高效、安全地管理虚拟组织。

2 问题及相关工作

虚拟组织需要一个访问控制模型来明确用户对资源的权限,而权限的赋予应该取决于上下文(时间、有效性等)、用户的身份以及所申请的资源。该模型还应在刻画实体的同时,保证虚拟组织中的动态实体无法随意破坏访问控制策略的有效性。此外,其管理模型应实现成员域管理自治。

现有的虚拟组织管理解决方案提供了多种访问控制模型。其中,以 DAC 及 MAC 为代表的静态、无约束访问控制模型不适合虚拟组织的动态特性。RBAC[2]引入了"角色"的概念,通过角色将主体与一个资源的访问许可相关联,用角色抽象出主体,但未对资源进行类似的抽象,也未消除资源的动态变化对虚拟组织访问控制的影响。此外,以上模型均没有处理上下文的能力。dRBAC[3]是一个 RBAC 模型的派生模型,该模型可以解决可升级的横跨多管理域的分布访问控制,但 dRBAC 仍然被与 RBAC 类似的弱点所困扰,无法很好地表达上下文约束以及对资源的抽象。文献[4]给出了基于信

*)基金项目:国家自然科学基金(90412007)。崔永瑞 博士研究生,研究方向为信息安全与网格计算;李明楚 教授,博士;胡红钢 博士。

任的虚拟组织访问控制模型，采用了划分信任等级的访问控制策略，可以描述虚拟组织成员之间更为丰富的信任关系，但未抽象出虚拟组织中实体的逻辑结构。文献[5]提出的访问控制模型实现了虚拟组织 partner 域的高度自治，但未描述上下文信息。文献[6]提出的 OrBAC 模型虽然可以表达上下文约束，但未刻画信任关系。为弥补上述模型的缺陷，本文给出了 TwBAC(Trustworthiness Based Access Control)模型。

3 TwBAC 模型

TwBAC 模型(如图 1(b))将具有相同特性的资源以及用户的行为分别抽象为“views”和“privileges”集合，并与“role”构成语义映射——“role 拥有对 view 的权限 privilege”。此外，TwBAC 抽象出上下文约束条件以及虚拟组织成员之间的信任度阈值的逻辑集合，通过定义实体集之间的关系来刻画带有上下文约束的访问控制策略以及虚拟组织成员之间的信任关系，弥补了 RBAC 模型(如图 1(a))的不足，此外，其管理模型 AdTwBAC，通过定义具有特殊语义的视图将管理职责分别交由各 partner 域来管理，既实现了 partner 域的管理自治，又减轻了虚拟组织的管理负担。

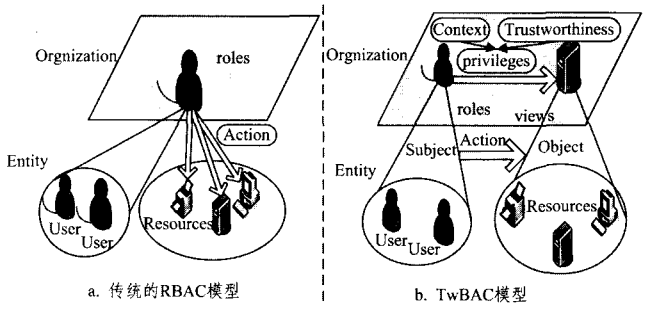

图 1 传统的 RBAC 模型与 TwBAC 访问控制模型的比较

3.1 TwBAC 模型的基本实体集与基本关系

TwBAC 有 9 类基本的实体集：*Org*(一个有组织的、在组织中扮演相应角色的主体团队)，*S*(一个 subject 集)，*A*(一个 action 集)，*O*(一个 object 集)，*R*(一个 role 集)，*P*(一个 privilege 集)，*V*(一个 view 集)，*C*(时间，空间等 context 集)，*T*(subject/role 与 object/view 之间的有向映射集，代表双方的信任度阈值)。其中，$Org \subseteq S, S \subseteq O$。即任何一个组织 *Org* 也是一个主体 *S*，任何一个主体 *S* 也是一个客体 *O*。而任何实体都拥有属性。例如，如果 *S* 是一个主体，那么 *name*(*S*)，*address*(*S*)分别代表它的名字和地址。

此外，还定义了这些集合之间的关系：

Empower 是一个在空间 $Org \times S \times R$ 上的关系，表示 *org* 赋予主体 *s* 以角色 *r*。

Use 是一个在空间 $Org \times O \times V$ 上的关系，*Use*(*org*,*o*,*v*)意味着 *org* 在视图 *v* 上使用客体 *o*。

Consider 是一个在空间 $Org \times A \times P$ 上的关系，意味着 *org* 认为 *action* α 属于权限 *p*。

TD 是一个在空间 $Org \times S \times O \times T$ 上的关系，代表着在 *org* 中，主体 *s* 对客体 *o* 的信任度阈值为 *t*。

Define 是一个在域 $Org \times S \times A \times O \times C \times T \times T$ 上的关系，代表着在 *org* 中，主体 *s*，*action* α，以及客体 *o* 之间存在着上下文约束 *c*，且 *s* 对 *o* 以及 *o* 对 *s* 的信任度阈值分别为 *tso*，*tos*。

Perm 是一个在空间 $Org \times R \times P \times V \times C \times T \times T$ 上的关系。*Perm*(*org*,*r*,*p*,*v*,*c*,t_{rv},t_{vr})意味着在上下文约束条件 *c* 下，当 *r* 对 *v* 的信任度大于 *trv*，且 *v* 对 *r* 的信任度大于 *tvr* 时，组织 *org* 赋予了角色 *r* 在视图 *v* 上执行权限 *p* 的许可。假定 $t_{rv}, t_{vr} \in [0,1]$。

Deleg 是一个在空间 $Org \times R \times P \times R \times C \times T$ 上的关系。*Deleg*(*org*,*r*1,*p*,*r*2,*c*,t_{rv})表示在上下文约束条件 *c* 下(通常为委托权限的时间限制)，当 *r*1 对 *r*2 的信任度大于 *trv* 时，在 *org* 中，角色 *r*1 将权限 *p* 委托给角色 *r*2。需要指出，这里的 *p* 可以是 *r*1 的全部或部分权限。事实上，*r*1 也可以对一个客体进行委托授权。

3.2 TwBAC 的管理模型(AdTwBAC)

网格虚拟组织可以看作一个在 TwBAC 上下文条件下的组织 *Org*。管理模型 AdTwBAC 实现组织的管理、角色的管理、权限的管理、视图的管理、上下文的管理等管理功能。它定义了特殊的管理视图，并通过向这些视图中添加客体来完成管理任务。这样，就可以将虚拟组织的管理职责分配给成员域来管理，实现成员域的管理自治。

3.2.1 URA(User Role Assignment)视图

URA 视图中的客体是一个组织用户与角色的映射，其含义为：“在特定上下文约束下为一个用户分配一个角色”。因此，为一个用户分配一个角色相当于在 URA(图 2)的视图上添加一个客体，这个客体应拥有 3 个属性：

—subject，代表等待分配角色的主体。

—role，代表为主体分配的角色。

—org，代表主体所在的组织。

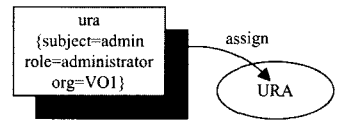

图 2 利用 URA 视图进行用户-角色映射

例如，为了在组织“Lab”的成员“lab1”中将一个特定的角色“professor”分配给一个用户，可以在 lab1 中生成一个视图“URA-prof-lab1”，它与 URA 的关系定义如下：

$$\forall ura, Use(Lab, ura, URA\text{-}prof\text{-}lab1) \Leftrightarrow$$
$$Use(Lab, ura, URA) \wedge (role(ura) = professor)$$
$$\wedge (org(ura) = lab1)$$

而“向视图 URA 添加客体”描述为：

$$\forall org, \forall ura, Use(org, ura, URA) \rightarrow$$
$$Empower(org(ura), subject(ura), role(ura))$$

假设“Assign”是管理员执行用户-角色分配时所使用的权限，即

$$Perm\begin{pmatrix} Lab, administrator, assign, \\ URA\text{-}prof\text{-}lab1, default, -1, -1 \end{pmatrix}$$

则可以将 *administrator* 角色赋予一个成员域的特定管理员，并赋予其分配与撤销用户的权限。

3.2.2 PRA(Permission Role Assignment)视图

为角色分配许可的过程与 URA 相似，可以向 PRA 视图中添加一个客体，该客体应有 7 个属性：

—issuer 代表发布此客体的组织。

—grantee，privilege，target 分别代表与分配的权限相关的 role，privilege 和 view。

—context 指明为角色分配许可的上下文。

—tdrv 与 tdvr 分别代表与此客体相关的 role 与 view 之

间的信任度阈值。

"向视图 PRA 添加客体"可描述为

$\forall pra, \forall org, Use(org, pra, PRA) \rightarrow$

$Perm(issuer(pra), grantee(pra), privilege(pra), target(pra), context(pra), tdrv(pra), tdvr(pra))$

3.2.3 VOA(View Object Assignment)视图

VOA 用来向资源视图中添加资源,同时可以将管理权赋予一个成员域的角色(设为 View-admin)。若 org 中的成员域"org2"提供存储资源,挑选其中的一个用户 dbadmin 管理本域资源。首先,赋予 dbadmin 以角色 View-admin:

$Empower(org, dbadmin, View\text{-}admin)$

然后,赋给 View-admin 权限"manage":

$$Perm\begin{pmatrix} org, View\text{-}admin, manage, \\ VOA\text{-}org2, default, -1, -1 \end{pmatrix}$$

其中,VOA-org2 是 org2 的资源视图,定义如下:

$\forall voa, Use(org, voa, VOA\text{-}org2) \Leftrightarrow$

$Use(org, voa, VOA) \wedge (org(voa) = org2)$

于是,在 org 中,将一个资源或客体加入到一个视图中等价于向 VOA-org2 视图中添加一个具有以下属性(资源或客体名称、视图名称、资源所在组织)的客体 voa,并由 view-admin 负责管理。

3.2.4 PDA(Privilege Delegation Assignment)视图

虚拟组织中的一个主体委托另一主体以相应权限,相当于在 PDA 中添加一个客体。该客体应有如下属性:

—issuer 代表发布此客体的组织。

—orig,subpriv,targ 分别代表与此次委托相关的源 subject/role,privilege 和目标 subject/role。

—context 指明委托的上下文。

—tdot 代表与此次委托相关的源 subject/role 对目标 subject/role 的信任度阈值。

"向视图 PDA 添加客体"描述为

$\forall pda, \forall org, Use(org, pda, PDA) \rightarrow$

$Deleg(issuer(pda), orig(pda), subpriv(pda), targ(pda), context(pda), tdot(pda))$

而 Deleg 与 Perm 有如下关系:

$\forall pda, Deleg(issuer(pda), orig(pda), subpriv(pda), targ(pda), context(pda), tdot(pda)) \Leftrightarrow$

$Perm(issuer(pda), tpda, subpriv(pda), view(opda), context(pda) \& context(opda), tdrv(opda), tdvr(view(opda)))$ $subpriv(pda) \subseteq privilege(opda) \wedge orig(pda) = opda \wedge targ(pda) = tpda$

这样,在 org 中,权限委托等价于向 PDA 视图中添加具有 issuer,orig,targ 等属性的客体 pda。并且,可以通过对信任度阈值 tdot 的设置来动态控制委托深度,避免委托的滥用。

3.3 TwBAC 的特点

TwBAC 模型较好地满足了虚拟组织的访问控制需求,能够刻画动态的、多管理域的网格虚拟组织环境,具有如下特点:

(1)TwBAC 通过资源的特性(如能力,QoS,配置等),抽象出资源逻辑层"view",并与"role"建立映射,形成访问控制策略,从而减轻管理负担。

(2)TwBAC 模型的管理模型 AdTwBAC 通过建立具有特殊管理语义的视图来为各成员域分配管理职责,实现管理自治。

(3)TwBAC 模型在刻画访问控制策略时引入信任度阈值,更好地刻画虚拟组织成员之间的信任关系。并且,在刻画委托授权策略时,引入信任度阈值参数,对委托深度进行控制。

(4)TwBAC 模型中加入了上下文语义,从而能够很好地刻画带有上下文约束的访问控制策略。

4 VO 建模实例

本节将以一个简单的实例来描述如何使用 TwBAC 模型为虚拟组织建模。如图 3,假设 Lab1 中的科学家需要大容量的存储设备来存储海量的实验数据,而 Lab2 中提供大容量的存储服务,他们需要联合起来形成一个虚拟组织(服务发现机制不在本文讨论范畴)。形成这样一个虚拟组织需要定义从两方面参与的实体,包括 roles,views,privileges 以及 administration authorities。根据 TwBAC 及 AdTwBAC 模型,将按照如下步骤对即将形成的虚拟组织建模:

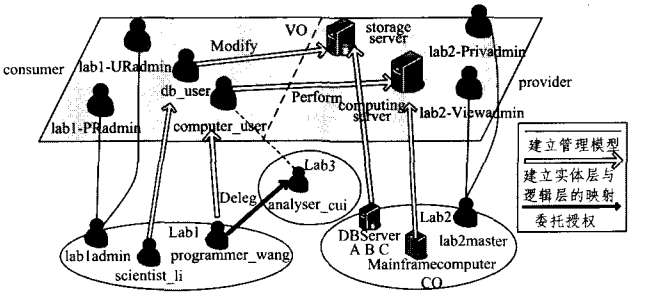

图 3 应用 TwBAC 建模实例

第 1 步 建立管理模型

先利用 AdTwBAC 模型建立虚拟组织的管理模型。虚拟组织本身作为 TwBAC 模型中的一个 org,应该拥有一些属性(如名字等)来将其与其它 VO 区分开。这里暂列 3 项必需:

$Name(VO) = collabration1$

$Partners(VO) = lab1, lab2$

$Time(VO) = timelimit$

首先,需要定义 collabration1 中的相关角色:

$$Relevant\text{-}role\begin{pmatrix} VO, lab1\text{-}URadmin, lab1\text{-}PRadmin, \\ lab2\text{-}Viewadmin, lab2\text{-}Privadmin \end{pmatrix}$$

同样,可以为 collabration1 定义相关视图和权限集。使用如下定义为 lab1 中的"lab1admin"赋予角色 lab1-URadmin,负责为 lab1 中的用户分配/撤销角色。

$Empower(VO, lab1admin, lab1\text{-}URadmin)$;

$$Perm\begin{pmatrix} VO, lab1\text{-}URadmin, manage, \\ URA\text{-}lab1, default, -1, -1 \end{pmatrix}$$

其中,URA-lab1 定义为

$\forall ura, Use(VO, ura, URA\text{-}lab1) \Leftrightarrow$

$Use(VO, ura, URA) \wedge org(ura) = lab1$

赋予"lab1admin"另一个管理角色"lab1-PRadmin",使其负责为 lab1 中的角色添加/删除权限:

$Empower(VO, lab1admin, lab1\text{-}PRadmin)$

$$Perm\begin{pmatrix} VO, lab1\text{-}PRadmin, manage, \\ PRA\text{-}lab1, default, -1, -1 \end{pmatrix}$$

其中,PRA-lab1 定义为

$\forall pra, Use(VO, pra, PRA\text{-}lab1) \Leftrightarrow$
$Use(VO, pra, PRA) \wedge grantee(pra) \subseteq$
$\{r / \exists ura \subseteq URA\text{-}lab1, role(ura) = r\}$
$\wedge privilege(pra) \subseteq privileges(VO)$

为 lab2 中的“lab2master”赋予角色“lab2-Viewadmin”，使其负责为 lab2 的客体分配/撤销视图，可以用以下定义：

$Empower(VO, lab2master, lab2\text{-}Viewadmin)$

$$Perm\begin{pmatrix} VO, lab2\text{-}Viewadmin, manage, \\ VOA\text{-}lab2, default, -1, -1 \end{pmatrix}$$

其中，VOA-lab2 定义为

$\forall voa, Use(VO, voa, VOA\text{-}lab2) \Leftrightarrow$
$Use(VO, voa, VOA) \wedge org(voa) =$
$lab2 \wedge Relevant\text{-}view(VO, view(voa))$

同样，赋予“lab2master”另一角色“lab2-Privadmin”，使其负责为 lab2 的 action 集添加/删除能够被 org2 理解的权限：

$Empower(VO, lab2master, lab2\text{-}Privadmin)$

$$Perm\begin{pmatrix} VO, lab2\text{-}Privadmin, manage, \\ AaA\text{-}lab2, default, -1, -1 \end{pmatrix}$$

其中，AaA-org2 定义为

$\forall aaa, Use(VO, aaa, AaA\text{-}lab2) \Leftrightarrow$
$Use(VO, aaa, AaA) \wedge org(aaa) = lab2$

第 2 步　建立实体层与逻辑层的映射

经过授权的角色/用户 lab1-URadmin 可以按照如下规则去为一个用户分配角色：

$Empower(VO, programmer\_wang, computer\_user)$
$Empower(VO, scientist\_li, db\_user)$

而 lab2-Viewadmin 可以在一个视图中使用一个客体：

$Use(VO, DBserverA, storageserver)$
$Use(VO, DBserverB, storageserver)$
$Use(VO, DBserverC, storageserver)$
$Use(VO, mainframecomputerCO, computingserver)$

lab2-Privadmin 可以为 VO 中的权限集添加实际的操作：

$Consider(VO, execute, Perform)$
$Consider(VO, read, Access)$
$Consider(VO, read\&write, Modify)$

lab1-RPadmin 可以允许一个角色在一个视图上拥有相应权限：

$$Perm\begin{pmatrix} VO, db\_user, Modify, \\ storageserver, workTime, 0.2, 0.8 \end{pmatrix}$$

$$Perm\begin{pmatrix} VO, computer\_user, Perform, \\ computingserver, dayTime, -1, 0.85 \end{pmatrix}$$

$$Perm\begin{pmatrix} VO, computer\_user, Access, \\ storageserver, workTime, -1, 0.8 \end{pmatrix}$$

在虚拟组织运行过程中，由于其动态特性，成员的改变可能会导致逻辑层的成员之间或逻辑层与实体层之间发生重新映射。例如，lab2 的计算资源因故撤销，lab2Viewadmin 需要重新向视图中加入资源。此时 computer_user/programmer_wang 对视图 computingserver 的信任度可能会随之改变，computer_user 需要根据策略决定是否继续使用此视图或选择其它视图。反之，如果 lab1 发生人事变动，由其他程序员来申请 computer_user 角色访问 computingserver，此时，computingserver 也可以根据其对 computer_user 信任度的变化决定是否提供服务。上述情况在虚拟组织运行过程中会经常出现，而本模型在应对时，并不需要调整逻辑层的安全策略，这将很大程度上减轻虚拟组织管理负担，提高运行效率。

第 3 步　委托(如有需要)

假设已经为 lab1admin 分配了角色 lab1-PDadmin 来管理 PDA，则如果“programmer_wang”在执行任务时需要委托 lab3 中的“analyser_cui”来访问 computingserver，lab1-PDadmin 可以实现此委托：

$$Deleg\begin{pmatrix} VO, computer\_user, Perform, \\ analyser, 12hours, 0.6 \end{pmatrix}$$

当然，同样的方法可以实现多级委托，并通过上下文约束及信任度阈值对委托深度进行控制。

这样，就使用 TwBAC 构建了网格虚拟组织模型，在实体层之上建立了统一的逻辑层，抽象出用户与资源。该框架允许不同的 partner 应用特定的访问控制策略参与其中，并保证对其自身成员管理的自治性与适应性。这个形式化描述应该配置在一个合作与协商的框架中来形成一个多管理的虚拟组织。

结束语　网格的动态性、多域性及异构性为虚拟组织的生成与管理带来了挑战。本文给出了一个基于信任度的虚拟组织访问控制模型 TwBAC。将虚拟组织中的用户和资源提供者抽象到统一的逻辑层上来，使其与底层域的具体管理机制分离，从而建立通用模型。该模型弥补了 RBAC 模型族在刻画虚拟组织方面的缺陷，有效地刻画了带有上下文约束的访问控制策略。其管理模型 AdTwBAC 实现了管理自治。此外，引入信任度的概念更准确地刻画了虚拟组织成员间的信任关系，并以此对委托深度进行控制。

下一步，将进一步完善 TwBAC 模型，对域内及域间成员间的信任度进行细粒度刻画，更好地描述虚拟组织成员间的信任关系。另外，将以 TwBAC 为基础，对 VO 的动态生成做进一步研究。

参 考 文 献

[1] Humphrey M, Thompson M R, Jackson K R. Security for grids // Proc. of IEEE. 2005, 93(3): 644-652

[2] Sandhu R S, Coyne E J, Feinstein H L, et al. Role-based access control models. IEEE Computer, 1996, 29(2): 38-47

[3] Freudenthal E, Pesin T, Port T, et al. dRBAC: distributed role-based access control for dynamic coalition environments // Proc. of 22nd International Conference on Distributed Computing Systems. 2002: 411-420

[4] Lin Aizhong, Vullings E, Dalziel J. A trust-based access control model for virtual organizations // Proc. of 5th International Conference on Grid and Cooperative Computing Workshops. 2006: 557-564

[5] 孙为群，单保华，等. 一种基于角色代理的服务网格虚拟组织访问控制模型. 计算机学报，2006，29(7)：1199-1208

[6] Kalam A A E, Baida R E, Balbiani P, et al. Organization based access control // Proc. of 4th IEEE International Workshop on Policies for Distributed Systems and Networks. 2003: 120-131