EDEMILSON DOS SANTOS DA SILVA

EXTENSÃO DO MODELO DE RESTRIÇÕES DO RBAC PARA SUPORTAR OBRIGAÇÕES DO MODELO ABC

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

CURITIBA

2004

EDEMILSON DOS SANTOS DA SILVA

EXTENSÃO DO MODELO DE RESTRIÇÕES DO RBAC PARA SUPORTAR OBRIGAÇÕES DO MODELO ABC

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: *Metodologias e Técnicas de Computação*

Orientador: Prof. Dr. Altair Olivo Santin

CURITIBA

2004

Silva,Edemilson dos Santos da
Extensão do Modelo de Restrições do RBAC para Suportar Obrigações do Modelo ABC. Curitiba, 2004. 90p.

Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

1. Segurança 2. Modelo de Restrições do RBAC 3. Modelo de Obrigações do ABC 4. Controle de Acesso Baseado em Quorum Mínimo. I.Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática Aplicada.

Esta página deve ser reservada à ata de defesa e termo de aprovação que serão fornecidos pela secretaria após a defesa da dissertação e efetuadas as correções solicitadas.

A minha esposa Giovania pelo amor e carinho, dedicação, compreensão e contribuição nos momentos mais difíceis.

Agradecimentos

Agradeço primeiramente a Deus pela vida e pelas oportunidades.

Agradeço à minha família pela força e incentivo.

Agradeço ao professor Altair Olivo Santin pela confiança.

Agradeço ao professor João Dias pelo apoio.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ACL	*Access Control List*
CORBA	*Commom Object Request Broker Architeture*
DAC	*Discretionary Access Control*
HTTP	*Hyper Text Transport Protocol*
IETF	*Internet Engineering Task Force*
LDAP	*Lightweith Directory Access Protocol*
MAC	*Mandatory Access Control*
OASIS	*Organization for the Advancement of Structured Information Standards*
OMA	*Object Management Architecture*
OMG	*Object Management Group*
ORB	*Objetct Request Broker*
RBAC	*Role-Based Access Control*
XACML	*eXtensible Access Control Markup Language*

Resumo

Este trabalho apresenta uma proposta de extensão ao modelo de restrição do modelo de controle de acesso baseado em papéis (*role-based access control* - RBAC) para suportar situações críticas respeitando as regras da política de autorização do sistema. As situações críticas exigem execução de atividades que não podem ser subdivididas em um conjunto de subtarefas a serem executadas seqüencialmente e nem podem ser realizadas por um usuário sozinho. A extensão proposta utiliza o conceito de obrigações do modelo ABC para permitir a criação de papéis (denominados papéis especiais) que só podem ser ativados com a autorização de um conjunto pré-definido de papéis simples (definidos pelo modelo RBAC). As atividades críticas são associadas aos papéis especiais. Ou seja, para que um principal realize uma atividade crítica, o mesmo deve ativar um papel especial que, por sua vez, necessita da aprovação de um quorum mínimo de principais para ser ativado. Os principais que participam da aprovação agem no sentido de aprovar a ativação do papel especial e não com o objetivo de exercer os direitos associados aos seus papéis.

Palavras-Chave: (Segurança, Modelo de Restrições do RBAC, Modelo de Obrigações do ABC, Controle de Acesso Baseado em Quorum Mínimo).

Abstract

This work presents an extension proposal to the model of restriction of the role-based access control (RBAC) model to support critical situations respecting the rules of the politics of authorization of the system. The critical situations demand execution of activities that cannot be subdivided set of sequential subtasks and a user alone cannot accomplish that one. The proposed extension uses the concept of obligations of the model ABC to allow the creation of roles (denominated special roles) that can only be activated with the authorization of a predefined group of simple roles (defined for RBAC model). The activities critics are associated to the special roles. In other words, so that a main one accomplishes a critical activity, the same should activate a special role that, for his time needs the approval of a minimum quorum of principals to be activated. The principals that participate in the approval act in the sense of approving the activation of the special roles and not with the objective of exercising the rights associated to their roles.

Keywords: (Security, Model of Constrained of RBAC, Model of Obligations of ABC, Control of Access based on Minimum Quorum)

Capítulo 1

Introdução

1.1. Desafio

Com o crescente uso das redes de computadores por organizações para conduzir seus negócios e o significativo aumento do uso da Internet, surgiu a necessidade de se utilizar melhores mecanismos para prover a segurança das transações de informações confidenciais. A questão segurança é preocupante quando imagina-se a possibilidade de ter informações expostas à sujeitos não autorizados, e que possuem meios cada vez mais sofisticados para violar a privacidade e a segurança das comunicações. Devido a estas preocupações a proteção da informação tem se tornado um dos interesses primordiais dos administradores de sistemas.

Uma das justificativas para o uso intensivo da informática como ferramenta para todas as áreas do conhecimento humano é o de facilitar o acesso de pessoas às informações. Porém, esta característica tem se tornado seu principal problema: a disseminação e o acesso indiscriminado e irrestrito a qualquer tipo de informação. Certamente não há um conflito, mas sim a necessidade de disciplinar o acesso através da criação e implantação de políticas com mecanismos de controle de acesso.

1.2. Motivação

A proteção da informação em um sistema computacional é definida pela política de segurança, um sistema é seguro se cumpre sua política de segurança. A política deve ser baseada no mundo real, definindo as propriedades que o sistema deve possuir para ser seguro e a responsabilidade das pessoas com a segurança.

Propriedades de segurança deve levar em conta três características importantes: confidencialidade, ou seja, só permitir acesso à informação para quem está autorizado para tal; integridade, para garantir a veracidade e a confiabilidade das informações; e disponibilidade de serviço para quem tem direito de acesso. Estas características podem ser alcançadas usando-se várias tecnologias, tais como: criptografia, assinatura digital, controle de acesso e modelos de segurança.

Os modelos de controle de acesso são classificados em duas categorias: discricionários e obrigatórios. Nos discricionários os acessos a cada recurso ou informação são manipulados sem restrição pelo proprietário ou responsável do sistema. Já nos obrigatórios (não-discricionários ou mandatórios) as autorizações de acesso são definidas através de um conjunto incontornável de regras que expressam algum tipo de organização envolvendo a segurança das informações no sistema como um todo [MAC97], baseando-se em uma administração centralizada de segurança, que dita quais serão as regras de acesso à informação.

O modelo discricionário é caracterizado pela sua flexibilidade o que facilita a gerência descentralizada. Ao contrário, os modelos obrigatórios exigem a presença no sistema de um administrador de segurança (*security officer*), que é assumido como único. O preço da flexibilidade do modelo discricionário é a falta de controle na discriminação da informação, pois são os próprios sujeitos que determinam quais os acessos que outros sujeitos possuem sobre seus objetos.

O RBAC é um esforço significativo no sentido de definir um modelo de controle de acesso que atenda de forma completa as necessidades de flexibilização e centralização da administração de segurança.

O controle de acesso baseado em papéis (RBAC) agregou as melhores características dos modelos clássicos (discricionário e obrigatório) no sentido de ser flexível como o primeiro e centralizado como o segundo, o que é difícil de se conseguir num mesmo modelo. Porém, o modelo de restrições do RBAC é baseado no princípio da separação de tarefas, que é sustentada pelo princípio do mínimo privilégio [SAN00]. Assim, a definição de políticas de autorização é fortemente influenciada pelo modelo de restrições. Outros tipos de restrições foram considerados e discutidos pelos estudiosos da área, mas não foram adotadas no modelo de referência.

A divisão de uma tarefa em subtarefas, executadas em seqüência, garante a conformidade com o modelo de restrição citado acima, mas restringe a especificação de políticas. Neste modelo não é possível a especificação de políticas em ambientes onde é necessária a concordância de um conjunto de principais para a realização de uma tarefa. Tarefas com este tipo de necessidade estão geralmente associadas a atividades não convencionais (críticas), por exemplo, envolvendo emergência médica ou um ambiente de segurança máxima.

Nos casos de emergências médicas, basicamente, há duas tendências com relação ao controle de acesso. Alguns autores baseiam suas propostas em fatores circunstanciais e outros no relaxamento do controle de acesso como alternativas para contornar as exigências específicas do ambiente.

1.3. Objetivos

Este trabalho tem por objetivo apresentar uma proposta de adequação do modelo RBAC com restrições para suportar atividades não convencionais respeitando as regras da política de autorização do sistema. Mais especificamente, deseja-se:

- Analisar os principais modelos de segurança computacional visando identificar modelos de restrições;
- Estudar detalhadamente os modelos de controle de acesso baseado em papéis;
- Analisar os modelo de restrições do RBAC e adequá-lo às necessidades de autorização baseada em quorum mínimo;
- Aplicar o modelo concebido ao ambiente médico de um hospital (setor de emergência);
- Analisar trabalhos que apresentam propostas de controle de acesso para situações de emergência médica;
- Construir um protótipo para validar o modelo proposto utilizando ferramentas padronizadas.

1.4. Organização

O trabalho apresentado nesta dissertação está estruturado em seis capítulos. Este capítulo inicial descreveu o contexto onde o trabalho está inserido, sua motivação e objetivos e um resumo dos seus principais resultados.

O capítulo 2 apresenta os fundamentos de um sistema de segurança computacional necessários ao entendimento deste trabalho, tais como políticas, violações, modelos e mecanismos de segurança.

O capítulo 3 apresenta um estudo mais detalhado sobre modelos de controle de acesso baseado em papéis. São analisados os principais modelos da literatura e algumas experiências de implementação.

O capítulo 4 apresenta uma análise detalhada dos modelos de restrições dos modelos baseados em papéis mais relevantes.

O capítulo 5 apresenta uma discussão detalhada sobre a proposta de extensão do modelo de restrições do RBAC. A capítulo 5 também discute aspectos relacionados à implementação, mostrando a estrutura do protótipo implementado, analisando os resultados obtidos, analisando um estudo de caso e discutindo melhorias para o protótipo.

Finalmente, o capítulo 6 apresenta as conclusões obtidas neste trabalho e possíveis trabalhos futuros.

Capítulo 2

Fundamentos de segurança em ambiente computacional

Nos últimos anos os sistemas computacionais de segurança tem sido alvo de muito interesse pela grande maioria das pessoas que se utilizam deles direta ou indiretamente. Na mídia, não são poucos os relatos encontrados sobre notícias de pessoas, instituições ou empresas que tiveram enormes prejuízos causados por ações intrusivas executadas pelos especialistas deste tipo de delito, que são conhecidos como sendo os *hackers*. Devido ao conhecimento de fatos como estes e muitas vezes à própria convivência com os mesmos é que chegamos à conclusão de que a principal finalidade da "segurança" consiste em promover a defesa de ataques externos como forma de evitar prejuízos de qualquer natureza que em sua grande parte geram perdas financeiras e até morais ao prejudicado.

O que iremos mostrar neste capítulo é que a segurança em sistemas computacionais trata-se de uma disciplina que busca através de todos os seus conceitos, metodologias e técnicas empregadas, manter as propriedades de um sistema de forma que ele não possa ser alvo de possíveis ações danosas praticadas por entidades não autorizadas junto às informações e recursos nele existentes. Existem hoje na literatura várias definições para segurança (*security*) e, em sua grande maioria, observamos a necessidade de se manter no sistema um conjunto de propriedades [DEN82]:

- A **confidencialidade**, a qual garante a revelação da informação somente a pessoas autorizadas.
- A **integridade**, que assegura a não modificação indevida – seja ela acidental ou intencional – da informação no sistema.

- A **disponibilidade**, que garante que as informações e recursos de um determinado sistema computacional estarão sempre desimpedidos e prontos para serem utilizados quando requisitados por pessoas autorizadas.

Neste capítulo iremos introduzir uma série de conceitos, modelos e técnicas utilizados com o objetivo de se manter as características acima mencionadas, garantindo assim a segurança da informação em sistemas computacionais.

2.1. Política de segurança

O termo Política de Segurança pode ter vários significados dependendo do nível em que é aplicado. Vendo sob o ponto de vista administrativo de uma instituição, podemos definir como sendo um conjunto de leis e práticas utilizadas pela instituição para gerenciar, proteger e distribuir suas informações. E esta mesma ótica administrativa em relação à segurança da informação deve-se refletir nos ambientes computacionais, onde a política de segurança passa a ser definida como sendo o conjunto de regras e serviços que visam especificar como um sistema provê os seus recursos mantendo sempre as propriedades de confidencialidade, integridade e disponibilidade. No texto que veremos a seguir, iremos nos ater à noção de política de segurança específica para ambientes computacionais.

As políticas de segurança são classificadas em duas categorias: as discricionárias e obrigatórias. Nas discricionárias os acessos a cada recurso ou informação são manipulados sem restrição pelo proprietário ou responsável do sistema, baseando-se na idéia de que este proprietário deve determinar quem tem acesso a estas informações segundo a sua vontade (à sua discrição). Já nas obrigatórias (não-discricionárias ou mandatórias) as autorizações de acesso são definidas através de um conjunto incontornável de regras que expressam algum tipo de organização envolvendo a segurança das informações no sistema como um todo [MAC97], baseando-se em uma administração centralizada de segurança, a qual dita qual serão as regras de acesso à informação.

2.2. Violações de segurança

As regras definidas pela política de segurança determinam as entidades que serão autorizadas e responsáveis pelas ações executadas sobre todas as informações mantidas no

sistema, as quais são normalmente identificadas como principais**.** Dependendo do nível de aplicação desta política pode-se caracterizar como sendo principal um usuário, um processo ou ainda uma máquina dentro de uma rede de computadores. Uma entidade caracterizada de usuário, processo ou máquina que ganha acesso a recursos de um sistema computacional de forma ilícita, violando assim a política de segurança é denominado de intruso.

As violações de segurança em sistemas computacionais se traduzem como sendo a arte de burlar de alguma forma a política de segurança de modo a passar despercebida uma ou mais das propriedades de segurança existentes. A tabela 2.1 mostra os tipos de violação alocados em contraposição às propriedades de segurança não verificadas [AMO94].

Tabela 2.1: Relação de violações as propriedades de segurança

Tipo de violação (TV)		**Propriedade de segurança violada**
I	revelação não autorizada	confidencialidade
II	modificação não autorizada	integridade
III	negação de serviço	disponibilidade

Antes de verificarmos as principais violações existentes no ambiente distribuído, devemos fazer algumas definições que serão importantes para o nosso entendimento. Uma *ameaça* (*threat*) é caracterizada como sendo uma ação possível que, uma vez concretizada, produz efeitos indesejáveis sobre os dados ou recursos de sistema. Uma ameaça quando posta em ação é identificada como um *ataque* (*attack*) à segurança do sistema. Entende-se por *vulnerabilidade* (*vulnerability*) como sendo uma falha ou característica indevida existente no sistema oriunda de falhas de concepção, implementação ou de configuração, que expõe os recursos deste sistema computacional a ataques e que podem ser exploradas para concretizar uma ameaça.

Uma maneira de ilustrarmos o relacionamento entre ameaças, vulnerabilidades e ataques é fazendo uma analogia com uma casa. Uma ameaça associada a uma casa é o roubo de móveis, dinheiro e eletrodomésticos. Vulnerabilidades podem ser comparadas a uma janela aberta ou uma porta que não esteja trancada. O ataque consiste na invasão propriamente dita com o conseqüente roubo dos bens existentes.

É muito comum encontrarmos na literatura o termo *misuse* que corresponde ao uso incorreto, impróprio ou excessivo de um recurso. A tabela 2.2 relaciona as ameaças mais

comumente presentes em relatos de violações de segurança em sistemas distribuídos [AMO94].

Tabela 2.2: Ameaças mais comuns ao sistema de Informática

Ameaça	TV	Descrição
Mascaramento (*masquerade/spoofing*)	II	Técnica utilizada para "se fazer passar"/forjar uma identidade. Utilizada por exemplo em capturadores de senha, *IP spoofing*, etc.
Bypassing control	I	Técnica utilizada para explorar vulnerabilidades do sistema. É comum a utilização desta técnica em invasões de sistemas com versões (*patchs, service packs*) desatualizadas.
Código maléfico (*malicious code*)	I II III	Técnica que utiliza software com código contendo partes aparentemente inofensivas ou invisíveis. Estas, quando executadas, comprometem a segurança dos recursos do sistema. Utilizada por exemplo em cavalos de tróia, vírus, bombas lógicas, etc.
Backdoor/Trapdoor	I II	Técnica utilizada para inserir/criar funções/escutas no sistema, que aceitam entradas específicas e permitem contornar/driblar os mecanismos de segurança do sistema.
Inspeção de lixeira (*midia scavenging*)	I	Técnica que consiste em bisbilhotar/revirar lixeiras procurando papéis, discos, etc, com informações importantes que não foram adequadamente destruídas.
Scanning	I	Técnica utilizada para bisbilhotar recursos do sistema com algum objetivo específico. Um ataque deste tipo utiliza-se de um software *scanner* para ser executado.
Negação de Serviço	III	Técnica utilizada para impedir o acesso legítimo ao sistema ou a algum recurso deste. Alguns exemplos desta técnica são *Distributed Denial of Service*, *Worms*, repetidor de discagem e diversas variações de *Denial of Service*.
Ataque nas Comunicações	I II	Técnica utilizada para escuta, interceptação, inserção ou alteração de mensagens durante a transmissão. Geralmente, a escuta da comunicação com um software *sniffer*, por exemplo, precede o uso de ataques como *Man-in-the-middle* ou roubo da sessão autenticada.

2.3. Modelos de segurança

Os modelos de segurança correspondem a descrições mais detalhadas do comportamento de um sistema, atuando sempre segundo regras de uma política de segurança estabelecida. Estes modelos são representados na forma de um conjunto de entidades e relacionamentos [GOG82]. Na literatura encontramos sempre os modelos divididos em três tipos básicos [SAN96, OSB00]:

- Baseados em identidade ou discricionários (*discretionary)*;
- Baseados em regras ou obrigatórios (*mandatory)*;
- Baseados em papéis (*roles*);

2.3.1. Modelos discricionários *(discretionary)*

Os controles discricionários, na sua essência, são baseados no modelo matriz de acesso proposta por Lampson [LAM71]. Neste modelo, o estado de proteção do sistema é representado através de uma matriz, onde as linhas correspondem aos sujeitos e as colunas correspondem aos objetos do sistema. Esta matriz relaciona-se através dos objetos, que podem ser definidos como sendo os recursos do sistema, dos sujeitos, que são as entidades ativas existentes no sistema e dos atributos de acesso, que são os direitos ou permissões de acesso (*read*, *write*) cabíveis ao sistema. Cada sujeito (S1, S2, S3, etc.) é representado por uma linha da matriz e cada objeto (O1, O2, etc.) por uma coluna. Na célula de intersecção das duas, onde ambas (linha x coluna) se encontram são especificados os atributos de acesso do respectivo sujeito em relação ao objeto considerado.

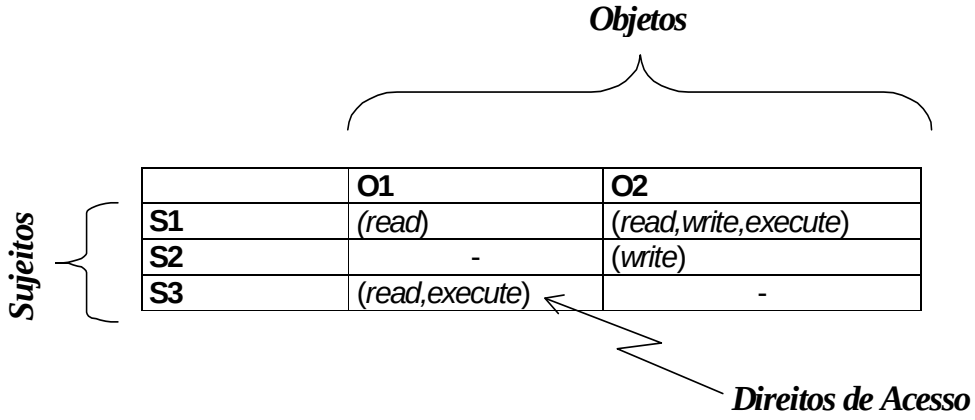


Figura 2.1: Modelo matriz de acesso

Se considerarmos uma coluna da matriz de acesso, veremos que a relação de todos os sujeitos existentes, com seus respectivos direitos de acesso sobre o objeto correspondem à coluna, formando o que é identificado como sendo uma lista de controle de acesso (*Access Control List* - ACL) do objeto considerado. As *ACLs* correspondem a uma forma de apresentação do modelo matriz de acesso. Na Figura 2.2, é apresentado um conjunto de *ACL*s onde cada lista corresponde a uma entrada do objeto correspondente.

Objeto	Listas de Controle de Acesso
O1	S1(read), S3(read, execute)
O2	S2(write), S1(*read*, *write*, *execute*)

Figura 2.2: Listas de Controle de Acesso

A ACL de um objeto permite uma fácil revisão dos acessos autorizados a um objeto. Outra operação bastante simples com o uso de ACLs é a revogação de todos os acessos a um objeto, bastando para isso substituir a ACL corrente por outra vazia. Por outro lado, a determinação dos acessos aos quais um sujeito está autorizado é bastante problemática; é necessário percorrer todas as ACLs do sistema para fazer este tipo de revisão de acesso. A revogação de todos os acessos de um sujeito também requer que todas as ACLs sejam inspecionadas e, eventualmente, modificadas.

A matriz de acesso pode ainda apresentar uma outra representação através de *Listas de Competência* (*capabilities list*). As *capabilities* são uma abordagem dual as listas de controle de acesso. Neste tipo de representação cada sujeito é associado a uma lista (a lista de capabilities) que indica, para cada objeto no sistema, os acessos que o sujeito está autorizado a efetuar. Isso corresponde a armazenar a matriz de acesso por linhas. Na Figura 2.3, cada lista (neste caso listas de *capabilities*) é representada em uma entrada por sujeito da matriz.

Sujeitos	Listas de Competências
S1	O1(read), O2(read, write, execute)
S2	O2(write)
S3	O1(read, execute)

Figura 2.3: Listas de Competências

As *capabilities* permitem uma fácil verificação e revogação dos acessos autorizados para um determinado sujeito. Porém, em contrapartida, a determinação de quais sujeitos poderão acessar um objeto requer a inspeção de todas as *capabilities* do sistema. As vantagens e desvantagens de *ACLs* e *capabilities* são, como as próprias estratégias, ortogonais entre si.

Capabilities são vantajosas em sistemas distribuídos. A posse de uma *capability* é suficiente para que um sujeito obtenha o acesso autorizado por esta *capability*. em um sistema distribuído, isso possibilita que um sujeito se autentique uma vez, obtenha a sua lista de

capability e apresente estas *capabilities* para obter os acessos aos quais ele está autorizado; os servidores precisam apenas verificar a validade dessa *capability* para liberar o acesso.

2.3.2. Modelos Obrigatórios (*mandatory*)

Os modelos obrigatórios que caracterizam as políticas obrigatórias baseiam-se em uma administração centralizada de segurança, a qual dita regras incontornáveis de acesso à informação. As mesmas definem regras e estruturas válidas no âmbito de um sistema, normalmente, especificando algum tipo de política multinível (*multilevel policy*). As políticas multiníveis estão baseadas em algum tipo de classificação ao qual estão submetidos os acessos dos sujeitos aos objetos.

Uma das formas de viabilizar a implementação das chamadas políticas multiníveis se dá com a construção de *reticulados* (*lattice*[1]) com *rótulos de segurança* (*security labels*). Os *rótulos de segurança* são constituídos por níveis de sensibilidade e categorias (*category*). As categorias correspondem a compartimentos específicos do sistema a que pertencem as informações em uma determinada instituição. Os níveis de sensibilidade atribuídos às informações no sistema derivam diretamente do tipo de classificação utilizado. Os rótulos de segurança correspondem ao produto vetorial do conjunto de níveis de sensibilidade pelo conjunto de *Category* (*Security label = Sensitivity level* x *Category*); sendo que *Category* indica o conjunto de todos os subconjuntos formados a partir das categorias definidas no modelo.

Assim sendo, pode-se definir que *Security label* é o conjunto de todos os pares ordenados (a,b) onde a, que corresponde ao primeiro elemento, está ligado ao nível de sensibilidade da informação no sistema e b descreve o subconjunto *category* (que pode até ser vazio) a que pertence esta mesma informação. Neste contexto o *sensitivity level*, que denota a sensibilidade do sujeito (entidade ativa no sistema) é denominado *autorização* ou *habilitação* (*clearance*) e do objeto (entidade passiva no sistema) é identificado como classificação (*classification*).

A seguir serão descritos os principais modelos obrigatórios constantes na literatura.

---

1 *Lattice* corresponde ao conjunto matemático de elementos parcialmente ordenados, os quais satisfazem uma relação que é transitiva e antissimétrica, de forma tal que para quaisquer 2 elementos, existe um elemento que é o maior no subconjunto de todos os elementos que são menores ou iguais a ambos, e um elemento que é o menor no subconjunto de todos os elementos que são maiores ou iguais a ambos.

2.4. O modelo Bell-LaPadula (BLP) de confidencialidade

Com o financiamento do governo dos Estados Unidos, no início da década de 70, diversas pesquisas sobre modelos de segurança e pretensão de violações de confiabilidade foram realizadas. Dois cientistas da MITRE Corporation, David Bell e Leonard LaPadula, desenvolveram um modelo baseado nos procedimentos usuais de manipulação de informações em áreas ligadas à segurança nacional americana. Esse modelo ficou conhecido como modelo Bell-LaPadula, ou o modelo BLP.

O modelo BLP [BEL73] classifica as informações em 4 níveis hierárquicos de sensibilidade (não classificada, confidencial, secreta e ultra-secreta – respectivamente do menor para o maior nível) e num conjunto não hierárquico de categorias ou compartimentos.

Ao utilizar este método de classificação reduz-se a complexidade das regras, aproximando o modelo BLP dos ambientes computacionais de uso corrente, sem enfraquecê-lo em nenhum aspecto.

O sistema é descrito em termos de sujeitos que acessam objetos, onde cada sujeito possui uma habilitação e cada objeto possui uma classificação. A cada sujeito está associado também um rótulo corrente de segurança - representando a classificação mais alta dentre as informações já consultadas pelo sujeito no sistema, sendo portanto, uma classificação flutuante (dinâmica). A habilitação de um sujeito sempre domina o seu rótulo corrente de segurança.

Para evitar a revelação da informação a sujeitos não autorizados, basicamente, o modelo BLP impõe duas regras:

- ***NRU*** (*No Read Up*) ou **propriedade de segurança simples** (*simple security propriety* ou *ss*): esta regra diz que um sujeito pode somente observar informações para as quais esteja habilitado, evitando assim que um sujeito de nível inferior leia informações de um nível mais elevado que o seu nível de segurança (habilitação e compartimento).
- ***NWD*** (*No Write Down*) ou **propriedade estrela** (**-propriety*): esta regra delimita um sujeito com *security label* x_s em seus acessos de escrita a objetos com nível de segurança x_o, onde x_o domina x_s.

O objetivo desta regra é evitar que a informação existente em um nível mais alto acabe passando para níveis mais baixos de segurança, o que seguramente caracterizaria a revelação não autorizada desta informação. Além das propriedades descritas e um conjunto de outras regras o modelo também mantém um controle discricionário por nível de segurança (*discretionary security propriety*), que reflete os princípios de autorização expressos no modelo matriz de acesso.

2.4.1. Dinâmica do Modelo Bell-LaPadula

Esta seção tem o propósito de analisar o comportamento dinâmico do modelo BLP, isto é, como o rótulo corrente de um sujeito evolui durante a operação do sistema.

Quando um usuário entra no sistema, ele recebe um rótulo corrente de segurança que seja dominado pela sua habilitação. Este rótulo pode ser escolhido pelo usuário ou atribuído automaticamente pelo sistema; a abordagem adotada não interfere no comportamento dinâmico. Os sujeitos criados em nome de um usuário herdam tanto a habilitação como o rótulo corrente de segurança do usuário. Os acessos destes sujeitos aos objetos devem sempre observar as propriedades ss e *.

Bell e LaPadula fornecem um conjunto de regras para a operação de um sistema seguro. Uma destas regras dita que o rótulo corrente de segurança de um sujeito somente é modificado mediante uma requisição explícita deste mesmo sujeito; isto significa que o rótulo corrente de segurança não flutua de maneira automática no sistema e também que esta flutuação ocorre sempre por iniciativa do próprio sujeito. A regra especifica que a alteração do rótulo corrente de segurança somente é autorizada se ela não violar a propriedade-*.

A dinâmica do modelo *BLP* pode ser resumidamente descrita por uma máquina de estados onde a transição de um estado seguro (condição em que nenhuma das propriedades de segurança é violada) para outro estado seguro se efetivará somente se as propriedades estrela e simples forem mantidas no acesso que define essa transição.

2.4.2. Limitações do Modelo Bell-LaPadula

O modelo de *BLP* apresenta entre as suas principais limitações [MCL90, MAC97], a escrita às cegas e a superclassificação da informação.

A Escrita às Cegas (*blind write*) é assim denominada porque a propriedade estrela permite que um sujeito escreva num objeto de nível de sensibilidade superior a sua

habilitação, o que pode determinar a destruição de informações sem caracterizar uma violação de segurança e das regras do modelo propriamente dito. O cenário de escritas cegas torna-se uma preocupação na medida em que o mesmo sujeito considerado inadequado para ver o conteúdo de um objeto possui permissão para fazer modificações arbitrárias neste mesmo objeto. Isto pode causar problemas de integridade que só podem ser resolvidos através de alterações nas regras do modelo BLP. Por exemplo, escritas em níveis de segurança mais altos que o corrente podem ser proibidas, ou seja, um sujeito somente poderia escrever em um objeto que tivesse o mesmo nível de segurança. Entretanto, tal modificação restringe, de certa forma, o modelo BLP e muda seu enfoque, que deixa de ser exclusivamente a ameaça de revelação não autorizada e passa a ser uma combinação de revelação e integridade. Por outro lado, a adoção de propriedade-* revisada é bastante comum em implementações de sistemas computacionais que seguem o modelo BLP.

A superclassificação *da informação* é determinada pelas regras do modelo que definem uma espécie de fluxo da informação dos níveis de segurança mais baixos para os mais altos o que dificulta, com o tempo, a manipulação da informação no sistema. Por exemplo, se um sujeito com rótulo corrente de segurança SECRETO deseja copiar um arquivo CONFIDENCIAL, a propriedade-* impõe que a cópia tenha classificação SECRETO, mesmo que as informações ali contidas possuam classificação CONFIDENCIAL. Ao longo do tempo, isso faz com que as informações subam no reticulado de rótulos de segurança, recebendo classificações sucessivamente maiores. A superclassificação da informação provoca a necessidade de reclassificações periódicas dos objetos (através de sujeitos de confiança) apenas para garantir a usabilidade de sistemas baseados no modelo BLP.

Na literatura existem várias técnicas propostas para tratar estes problemas do BLP [MCL90, MAC97]. Dentre elas podemos citar os *sujeitos de confiança* (*trusted subject*), o qual foi considerado já em [BEL73], para que os processos possam violar a propriedade estrela, se necessário, sem comprometer a segurança do sistema, tratando entre outras coisas a superclassificação da informação. Como exemplo, podemos citar o conceito de confiança como sendo utilizado para qualificar os processos relacionados com a manutenção do sistema, pois se o administrador do sistema tiver que obedecer estritamente às regras do modelo BLP ele dificilmente conseguirá realizar qualquer tarefa significativa de administração. Outra classe de processos que faz uso da noção de sujeitos de confiança compreende os subsistemas mais críticos do sistema operacional, como gerência de memória e *drivers* de dispositivos.

2.5. O modelo Biba de Integridade

O modelo Bell-Lapadula [BEL73] tem por principal objetivo conter ameaças de revelação não autorizada; não obstante, os próprios criadores do modelo BLP discutem como ele poderia ser adaptado para conter as ameaças de integridade. Embora as idéias de Bell e LaPadula necessitem de uma maior consistência, elas serviram muito bem de base para que Biba, também integrante da MITRE Corporation, desenvolvesse, na segunda metade da década de 70, um modelo de segurança com o propósito de garantir a integridade da informação. Este modelo ficou conhecido como sendo o Modelo de Integridade Biba, ou simplesmente chamado de modelo Biba [BIB77].

O modelo Biba é definido como o dual do BLP. Suas regras são similares às do modelo anterior, porém, tem como objetivo, a preservação da integridade das informações classificadas, evitando alterações não autorizadas.

O modelo define níveis hierárquicos de integridade para os sujeitos (I_s) e para os objetos (I_o) similares aos níveis de sensibilidade definidos no BLP. A propriedade simples de integridade, também conhecida como propriedade-is ou regra *no read down* (NRD), define que um sujeito só pode ler um objeto se seu nível de integridade for dominado pelo do objeto ($I_o \geq I_s$). A propriedade estrela de integridade especifica que um sujeito pode ter direito de escrita sobre um objeto, se e somente se $I_s \geq I_o$.

Por ser o dual do modelo BLP, este modelo apresenta limitações similares às descritas naquele. No modelo Biba ocorre uma degradação do nível de integridade, de maneira análoga à superclassificação da informação do modelo de BLP. Via de regra, também há a necessidade de sujeitos de confiança no modelo Biba para alterar a integridade de sujeitos e objetos, mantendo o sistema viável.

2.5.1. Limitações do Modelo Biba

O modelo Biba possui muitas das vantagens e limitações do modelo BLP devido à sua similaridade com o mesmo. Por exemplo, ambos os modelos são relativamente simples e intuitivos, e conseguem demonstrar de uma maneira bastante intuitiva a sua capacidade de conservar a propriedade de segurança considerada (confidencialidade no BLP e integridade no Biba).

Entretanto, foi sugerido que o modelo Biba, assim como o BLP, depende em demasia de sujeitos de confiança em situações práticas: a necessidade de um processo de confiança para aumentar ou reduzir a integridade de sujeitos ou objetos é especialmente problemática para a integridade. Outra crítica ao modelo Biba é a ausência de provisão de mecanismos para a promoção da integridade de um sujeito ou objeto. Cabe notar que todas as mudanças possíveis no modelo Biba preservam a integridade de todos os sujeitos ou rebaixam a integridade de algum sujeito ou objeto. Isso permite imaginar que com o passar do tempo os sistemas sofrem um rebaixamento do nível de integridade, monotonicamente decrescente com os sujeitos e objetos migrando gradativamente para o nível mais baixo de integridade. Esta degradação da integridade da informação é análoga ao problema de superclassificação da informação no modelo BLP.

A implicação, presente no modelo Biba, de que integridade é uma medida e que a noção de "maior integridade" deve ter algum significado é criticada por muitos dos pesquisadores existentes. O seu argumento é que a integridade de sujeitos e objetos deve ser encarada como um atributo binário, que simplesmente está ou não presente.

2.6. Outros Modelos Obrigatórios

Na literatura são identificados outros modelos obrigatórios além do Bell-Lapadula e do Biba. O modelo Clark-Wilson (CW) [CLA87], por exemplo, foi criado em 1987 por David Clark, do MIT e David Wilson, da ERNEST AND WHINNEY. Este novo modelo foi motivado principalmente pela maneira como organizações comerciais controlam a integridade de seus documentos em papel em um ambiente de trabalho não-automatizado. Em outras palavras Clark e Wilson consideraram práticas administrativas e contábeis largamente difundidas e tentaram extrapolá-las para uso em aplicações computacionais. O modelo de integridade resultante deste esforço tem sido considerado bastante efetivo como guia para projetistas e desenvolvedores de sistemas computacionais onde a integridade desempenha um papel importante.

O modelo Clark-Wilson é baseado na idéia que a integridade é mais importante que a confidencialidade [CLA87] para operações comerciais. Mas diferente dos modelos Bell-LaPadula e Biba, o CW assume transações bem-formadas ("todos os passos de uma seqüência de atividades são executados corretamente") e a separação de tarefas ("cada sujeito

desempenha um papel distinto na seqüência de atividades que formam uma transação") como essência de sua definição.

Assim como nos outros modelos o CW também possui algumas limitações. A principal delas é que os procedimentos de verificação de integridade e as técnicas para garantir que os procedimentos de transformações preservem a integridade não são facilmente implementadas em sistemas computacionais. É perfeitamente possível conceber a sua implementação em aplicações restritas, mas em aplicações menos triviais o uso de procedimentos de verificação de integridade e procedimentos de transformações é muito mais complexo. Outro aspecto problemático do modelo Clark-Wilson é o impacto no desempenho do sistema causado pela implementação das triplas-CW (triplas de acesso do tipo sujeito/programa/dados).

Outra classe de modelos obrigatórios presente na literatura é a dos modelos de Controle de Fluxo, introduzidos no sentido de tratar os problemas de confinamento (canais cobertos temporais, canais de memória, cavalos de Tróia, etc.). Nestes modelos não são considerados mais acessos de um sujeito em objetos, mas de fluxos de informação entre sujeitos. Estes modelos tratam de identificar os canais de comunicação legítimos e os ilegítimos. O modelo descrito em [DEN76] é um exemplo desta classe de modelos.

2.7. Modelos Baseados em Papéis (Role-Based Access Control - RBAC)

Os modelos Baseados em Papéis (RBAC) tem como objetivo intermediar o acesso dos usuários à informação com base nas atividades que são por eles desenvolvidas no sistema. A idéia central é que o usuário desempenhe diferentes papéis (*roles*) em um sistema. Um papel pode ser definido como um conjunto de atividades e responsabilidades associadas a um determinado cargo ou função em uma organização. Assim, no RBAC, as permissões são conferidas aos papéis e os usuários são autorizados a exercer papéis (Figura 2.4). O controle de acesso baseado em papéis facilita a gerência de autorização, porque quando o usuário muda de atribuição – sendo, portanto, desassociado de um papel e assumindo um outro – a manutenção das permissões dos papéis não sofre mudanças. Em geral, o RBAC trabalha com o princípio do mínimo privilégio, um usuário ativa apenas o subconjunto de papéis que precisa para executar uma operação, esta ativação pode ou não estar sujeita a restrições.

O RBAC não é um conceito novo, mas só recentemente vem ganhando a atenção dos pesquisadores. Um modelo unificado denominado RBAC-NIST foi criado para tentar

padronizar as várias tendências que têm surgido em modelos de papéis [SAN00]. Esta família de modelos RBAC-NIST se apresenta estratificada a partir do modelo RBAC básico (*Flat RBAC*), evoluindo pelos outros modelos da família – RBAC Hierárquico (*Hierarchical RBAC*), RBAC com Restrições (*Constrained RBAC*) e RBAC Simétrico (*Symmetric RBAC*) – adicionando funcionalidades ao mesmo.

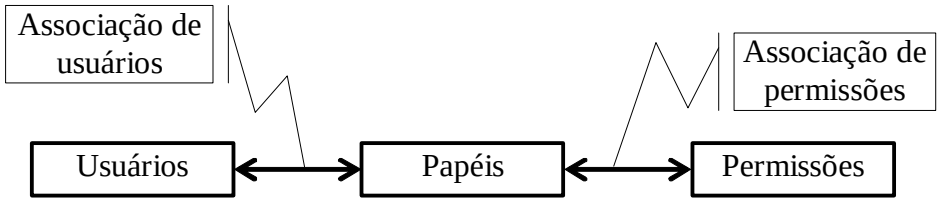


Figura 2.4: Modelo RBAC Básico

O RBAC básico [SAN00] implementa a infra-estrutura básica composta pelo usuário, permissões e papéis com suas respectivas semânticas. A relação entre os mesmos deve ser do tipo muitos-para-muitos (Figura 2.4). Este modelo define um suporte à revisão de associações usuário-papel, o que determina os usuários associados a um papel e vice-versa.

O RBAC hierárquico [SAN00] acrescenta ao RBAC básico uma hierarquia que permite estruturar papéis de maneira a refletir as responsabilidades de uma organização real.

O RBAC com Restrições [SAN00] está calcado no princípio do mínimo privilégio (suporta a *separação de tarefas*: o usuário só utiliza os direitos necessários para uma dada tarefa), impondo desta forma restrições na ativação de papéis.

O RBAC Simétrico [SAN00] inclui suporte à revisão de associações permissão-papel, o que torna possível identificar as permissões associadas a um papel e os papéis que possuem determinadas permissões.

O RBAC será discutido em maiores detalhes no capítulo 3.

2.8. Considerações sobre os modelos

Basicamente os modelos de segurança foram definidos em períodos distintos com tendências e características específicas. No começo dos anos 60 estimulados pelo desenvolvimento dos sistemas de tempo compartilhado. Depois, nos anos 70, com fins e propósitos militares, e nos anos 80 com enfoque mais comercial. Dos anos 90 em diante, a

tônica está sendo em modelos que envolvem os ambientes distribuídos, sem um propósito específico, mas geralmente para suportar políticas múltiplas. A diversidade de origem e objetivos dos modelos de segurança não permite que eles sejam diretamente comparados entre si. Entretanto, é possível abstrair-se de detalhes específicos de cada modelo e concentrar-se nas suas características gerais, estabelecendo uma base comum a partir da qual é possível tecer uma comparação. A tabela 2.3 ilustra as principais diferenças entre os modelos de controle de acesso discutidos neste capítulo.

O modelo matriz de acesso é caracterizado pela sua flexibilidade o que facilita a gerência descentralizada. Como neste modelo são os próprios sujeitos que determinam quais os acessos que outros sujeitos possuem sobre seus objetos, diz-se que a administração da política de segurança do modelo matriz de acesso é descentralizada. Ao contrário, os modelos obrigatórios exigem a presença no sistema de um administrador de segurança (*security officer*), que é assumido como único.

O preço da flexibilidade e da descentralização é a complexidade envolvida no controle à propagação de direitos no sistema. Os modelos obrigatórios geralmente definem um conjunto de regras não contornáveis no sistema que diminui as possibilidades de propagação de direitos. Estes modelos são próprios para gerências centralizadas de segurança, representando estruturas pouco flexíveis. Os modelos baseados em papéis são tidos como modelos intermediários entre os discricionários e os obrigatórios. Apresentam a estrutura flexível dos primeiros e gerência que pode ser centralizada dos obrigatórios. Todos os modelos descritos neste texto podem ser implementados em sistemas distribuídos, com maior ou menor dificuldade [WES00].

A separação de tarefas é um conceito suportado somente pelos modelos Clark-Wilson ou baseados em papéis. O princípio do mínimo privilégio, por sua vez, é suportado tanto pelos modelos baseados em papéis quanto pelos modelos BLP e Biba, através do uso criterioso dos rótulos correntes de segurança. É importante observar, porém, que a granularidade do suporte a mínimo privilégio nos modelos baseados em papéis é bem mais fina do que nos modelos baseados em rótulos. Os modelos baseados em papéis são os únicos que permitem incorporar, de alguma forma, a hierarquia natural das organizações ao controle de acesso; nenhum dos outros modelos fornece esta importante facilidade ao administrador de segurança. O último critério considerado na comparação da tabela é a facilidade ou viabilidade de implementação. O modelo matriz de acesso é o mais simples de ser implementado dentre todos. Modelos

baseados em papéis vêm em segundo lugar em termos de simplicidade, não trazendo grandes complicações à implementação. Os modelos baseados em rótulos já foram implementados em vários sistemas, mas requerem um razoável esforço para identificar todas as estruturas de um sistema que precisam ser rotuladas para evitar as violações de confiabilidade (BLP) ou integridade (Biba). O modelo Clark-Wilson é o mais complexo em termos de implementação uma vez que sua construção e certificação de procedimentos de verificação de integridade e procedimentos de transformações é uma tarefa muito difícil em sistemas reais.

Tabela 2.3: Comparação entre os modelos de segurança

Características	**Modelo**				
	Matriz de Acesso	**BLP**	**Biba**	**CW**	**Modelo de Papéis**
Flexibilidade	Sim	Não	Não	Não	Sim
Política de Controle de Acesso Centralizada	Não	Sim	Sim	Sim	Sim
Administração da Política Centralizada	Não	Sim	Sim	Sim	Sim ou Não
Separação de Tarefas	Não	Não	Não	Sim	Sim
Suporte a Mínimo Privilégio	Não	Sim	Sim	Não	Sim
Suporte à Hierarquia Organizacional	Não	Não	Não	Não	Sim
Facilidade de Implementação	Fácil	Médio	Médio	Difícil	Fácil

2.9. Mecanismos de segurança

Os mecanismos são responsáveis pela implementação das políticas de segurança específicas, expressadas pelos modelos de segurança. Como exemplo podemos dizer que uma política de segurança pode exigir que todos os usuários de um sistema sejam identificados univocamente para fins de contabilidade, e os mecanismos para implantar esta política incluem o uso de senhas, de cartões magnéticos e de dispositivos de reconhecimento de

impressões digitais. Para viabilizar a implantação de tais políticas, os mecanismos são construídos a partir de controles de acesso e controles criptográficos.

No que se refere a controle de acesso, podemos dizer que virtualmente todos os sistemas computacionais podem ser descritos em termos de sujeitos acessando objetos. O controle de acesso é, portanto, a mediação das requisições de acesso a objetos iniciados pelos sujeitos. *Um monitor de referência* é um modelo conceitual do subsistema responsável pelo controle de acesso; é a entidade que recebe todas as requisições de acesso dos sujeitos e autoriza ou nega o acesso de acordo com a política de segurança implantada.

O monitor de referência, tendo como função intermediar todas as requisições de acesso aos objetos de um sistema deve ter algumas propriedades: deve ser inviolável, incontornável (sempre invocado) e pequeno o suficiente para permitir a verificação de sua correção. A noção de núcleo de segurança foi definida em [LAN83] como o conjunto de recursos de hardware e software que permitem a realização de um monitor de referências.

A implementação do monitor de referência é realizada através de mecanismos de controle, que podem ser discricionários (DAC – *Discretionary Access Control*), obrigatórios (MAC – *Mandatory Access Control*) ou baseados em papéis (*RBAC - Role-Based Access Control*).

2.9.1. Mecanismos DAC (*Discretionary Access Control*)

Os mecanismos DAC implementam políticas discricionárias, permitido ao usuário atribuir direitos de acesso sobre seus recursos computacionais de acordo com a sua vontade. Baseia-se na idéia de que o proprietário da informação deve determinar quem terá acesso a essa informação. O controle discricionário permite que os dados sejam livremente copiados de objeto para objeto, de modo que, mesmo que o acesso aos dados originais seja negado, pode-se obter acesso a uma cópia. Porém, se o usuário não atribuir corretamente estes direitos, ou mesmo se o fizer permitindo acesso de cópia a outros sujeitos, a disseminação no sistema de suas informações não pode ser controlada. O controle de acesso discricionário não impõe nenhuma restrição à disseminação de direitos e à própria evolução da matriz de acesso. Na prática, devido talvez à facilidade de implementação, o controle de acesso discricionário é largamente utilizado nos sistemas atuais.

2.9.2. Mecanismos MAC (*Mandatory Access Control*)

Em mecanismos MAC, que implementam a política obrigatória, as regras de controle de acesso são impostas por uma autoridade central. Baseia-se em uma administração centralizada de segurança, a qual dita regras incontornáveis de acesso à informação. Estes mecanismos, como descrito anteriormente, implementam políticas multinível. Dos vários relatos de implementação de mecanismos MAC, observa-se que os mesmos são bem mais difíceis de viabilizar que os de DAC, devido à rigidez de suas regras e às limitações de seus modelos, além de outras dificuldades de caráter computacional [LAN84]. A forma mais usual de controle de acesso obrigatório é o controle de acesso baseado em reticulados (*lattice-based access control*), que confina a transferência de informação a uma direção em um reticulado de rótulos de segurança.

2.9.3. Role-Based Access Control

Para os mecanismos que implementam RBAC a identidade no sistema é o papel, uma vez que estes encapsulam as políticas na forma de permissões. Tem como base que os direitos de acesso sejam atribuídos a papéis e não a usuários, assim como acontece no DAC, já que os usuários obtêm estes direitos em virtude de terem papéis a si atribuídos. Por ser independente das políticas, o RBAC é facilmente ajustável a mudanças no ambiente e deve ser largamente utilizado, porque não é tão flexível quando o DAC e nem tão rígido quanto o MAC. Por mais que os modelos RBAC ainda estejam em desenvolvimento, existem vários relatos de implementações do mesmo, um deste pode ser encontrado em [GLE99].

2.10. Controles Adicionais de Segurança

Os mecanismos de segurança têm como sue principal objetivo implantar políticas de segurança, envolvendo para isso ações, técnicas ou dispositivos. Mas indo um pouco além, existem ainda outros controles (internos) que não atuam diretamente nas requisições de acesso mas que devem necessariamente estar presentes nos sistemas. Dentre elas podemos citar:

- A auditoria de vestígio, ligada à geração periódica de registros de eventos associados à segurança, coletados para uso potencial em detecção de intrusão e/ou auditoria de segurança.

- A auditoria de segurança, inspeção independente (por terceiros) dos procedimentos e registros do sistema com intuito de verificar adequação da política de segurança e as possíveis violações do sistema.
- A detecção de intrusão, usa os mesmos registros das auditorias em métodos automatizados de análise em tempo real, envolvendo muitas vezes uma seqüência de eventos relacionados ou não, com o intuito de identificar atividades anormais no sistema.

Mecanismos que façam uso de técnicas de *backups*, replicações e que permitam recuperar o sistema em situações onde as violações (seção 2.2) não puderam ser evitadas completam os controles adicionais.

Outros controles (externos) necessitam ser adicionados aos internos já comentados. Por exemplo, é necessário que se leve ao conhecimento de cada usuário suas atribuições e responsabilidades, para que esse saiba o que está autorizado a fazer. Se este não estiver treinado e convencido da importância da segurança no ambiente computacional as demais medidas podem se tornar sem eficácia [AMO94].

2.11. Conclusões do Capítulo

Os Modelos de Segurança são uma importante ferramenta para a definição de políticas de segurança para qualquer ambiente computacional. Ao longo dos anos, diversos modelos de segurança foram propostos, baseando-se nas mais variadas premissas e visando atender a diferentes aspectos de segurança.

Este capítulo procurou fazer um apanhado dos trabalhos mais recentes na área de modelos de segurança computacional, destacando modelos considerados clássicos na área e que ainda hoje são largamente utilizados nos mais variados ambientes. Neste capítulo vimos a essência destes modelos, analisando-se criticamente a sua validade e adequação, bem como alguns aspectos de sua implementação.

Os modelos baseados em papéis possuem diversas vantagens sobre os outros modelos, entre as quais podemos destacar a separação de tarefas, o princípio do mínimo privilégio e a possibilidade de incorporar, de alguma forma, a hierarquia natural das organizações ao controle de acesso; nenhum dos outros modelos fornece esta importante facilidade ao administrador de segurança. Devido às suas características os sistemas baseados em papéis são uma importante tendência na área de controle de acesso em sistemas computacionais.

Capítulo 3

Controle de Acesso Baseado em Papéis (RBAC – *Role-Based Access Control*)

O controle de acesso baseado em papéis (RBAC - *role-based access control*) é um conceito que surgiu com os primeiros sistemas computacionais interativos de múltiplos usuários e múltiplas aplicações, no início da década de 70. A principal característica do RBAC é associar as permissões de acesso a papéis (*roles*), e estes papéis são associados a usuários. Papéis são criados de acordo com os diferentes cargos ou funções em uma organização, e os usuários são associados a papéis de acordo com suas responsabilidades e qualificações. Os usuários podem ser facilmente remanejados de um papel para outro. Mudanças no ambiente computacional, como instalação de novos sistemas ou remoção de aplicações antigas, modificam o conjunto de permissões atribuídas aos diferentes papéis [SAN96].

Este capítulo introduz alguns conceitos sobre RBAC, apresenta detalhadamente o modelo unificado RBAC-NIST, discute outros modelos considerados relevantes e apresenta algumas implementações de RBAC.

3.1. Introdução

Mesmo estando presente sob diversas formas nas últimas décadas somente Ferraiolo em 1992 [FER92] propôs um modelo baseado no conceito RBAC, porém o modelo começou a ser formalizado em 1995 com a adesão de vários pesquisadores da área ao modelo [FER95, GAV98, KUH97].

Ferraiolo [FER95] acredita que as principais motivações do RBAC são, em primeiro lugar, a capacidade para expressar e impor uma política de segurança específica para uma organização e, em segundo lugar, para simplificar o oneroso processo de gerenciamento de segurança. O modelo RBAC permite definir políticas voltadas às reais necessidades das instituições através de características que tornam a segurança independente da vontade do usuário, simplificando o gerenciamento de autorizações e tornando mais flexível a especificação e a imposição de políticas de proteção. A enumeração de todas as funções (papéis) existentes dentro de uma instituição e a definição de regras de acessibilidade aos objetos necessários para o exercício de cada função é a base do modelo. Ao vincular os usuários aos papéis, os mesmos herdam automaticamente os direitos de acesso do papel a que pertencem. Desta maneira, a administração da política de segurança resume-se à definição de quais direitos de acessibilidade aos objetos são necessários para o exercício de cada função dentro da organização. O modelo ainda suporta que um papel poderá ser formado a partir da combinação de outros. Desta maneira o gerenciamento das permissões de acesso dos usuários é uma tarefa impessoal, flexível, descentralizada e simplificada.

3.2. As variações do Modelo RBAC

Após o artigo de Ferraiolo [FER92] que, através da argumentação, propunha uma nova política voltada à realidade da estrutura das organizações civis, porém, sem uma especificação formal que fechasse o modelo, surgiram nos anos seguintes várias propostas e implementações de modelos [FER95, FER99, GUI95, NYA99, CHA98, SAN96, SAN98a, SAN98b]. Estas propostas traziam variações de modelos RBAC, mas todos girando em torno do rigor de ter-se papéis, elementos aos quais são atribuídos direitos de acesso.

Os modelos foram propostos e implementados de forma independente sem qualquer esforço para padronização do RBAC. Para promover o desenvolvimento e o uso da tecnologia de controle de acesso baseado em papéis é necessário o estabelecimento de padrões. O primeiro passo para a padronização foi dado pelo NIST com a apresentação do modelo unificado RBAC-NIST [SAN00].

3.3. O Modelo Unificado RBAC-NIST

O primeiro modelo RBAC formalizado na literatura foi o de Ferraiolo e Kuhn [FER92], do NIST. O trabalho de Sandhu et. all. [SAN96] foi o primeiro a reconhecer a

impossibilidade de agregar todas as nuances do RBAC em um único modelo, o que levou à definição da família RBAC96 de modelos. O modelo original do NIST também passou por revisões, e alguns de seus conceitos foram atualizados ao longo do tempo [FER95, FER99]. Todos estes modelos compartilham um núcleo comum de conceitos, mas cada um deles possui particularidades que os diferenciam. Entretanto, há visivelmente bem mais semelhanças do que diferenças. Isto levou o NIST e o grupo liderado por Sandhu a propor um modelo unificado que padronizasse os conceitos do RBAC, numa tentativa de estabelecer um consenso e um ponto de partida para novos desenvolvimentos.

Este modelo, baseado largamente na família RBAC96 e no modelo NIST, ficou conhecido como modelo unificado RBAC-NIST [SAN00]. Embora alguns aspectos do modelo unificado tenham sido contestados [JAE00], a proposta foi bem recebida pela comunidade.

3.3.1. Estrutura do Modelo

Sendo o RBAC um conceito amplo e aberto existe um consenso de que um modelo único definitivo para o RBAC estaria fora da realidade, por que seria excessivamente restritivo ou complexo, representando apenas um dos pontos possíveis dentro do espectro [SAN96, SAN00]. O modelo RBAC-NIST define uma família de modelos, que parte de um componente básico contemplando as características fundamentais do RBAC passando por componentes adicionais que acrescentam funcionalidade e requisitos ao modelo básico. A família de modelos RBAC96 [SAN98b, SAN96] é, provavelmente, o exemplo mais conhecido deste tipo de abordagem.

A família RBAC-NIST, como comentado no capítulo anterior, define quatro modelos:

- RBAC Básico (*Flat RBAC*)
- RBAC Hierárquico (*Hierarchical RBAC*)
- RBAC com Restrições (*Constrained RBAC*)
- RBAC Simétrico (*Symmetric RBAC*)

A hierarquização do modelo RBAC-NIST pode ser interpretada de duas maneiras. Na interpretação original, a ordenação dos modelos é seqüencial e incremental, ou seja, cada modelo possui todas as características do anterior e ainda características adicionais. Nesta

abordagem existem sete configurações possíveis que podem ser visualizadas na Tabela 3.1. A interpretação predominante (mais flexível) do modelo reconhece o RBAC Básico e o RBAC Hierárquico como uma seqüência, mas trata o RBAC com Restrições e o RBAC Simétrico como requisitos independentes, não ordenados.

Tabela 3.1: Configurações possíveis na interpretação ordenada dos modelos

Nível	**Modelo**	**Restrições**	**Revisão Permissão-Papel**
1	Básico	Não	Não
2a	Hierárquico Geral	Não	Não
2b	Hierárquico Limitado	Não	Não
3a	com Restrições Hierárquico Geral	Sim	Não
3b	com Restrições Hierárquico Limitado	Sim	Não
4a	Simétrico Hierárquico Geral	Sim	Sim
4b	Simétrico Hierárquico Limitado	Sim	Sim

A interpretação não-ordenada permite, além das configurações da Tabela 3.1, cinco configurações adicionais (Tabela 3.2), o que reflete a superioridade desta abordagem.

Tabela 3.2: Configurações adicionais na interpretação não ordenada dos modelos

Modelo	**Restrições**	**Revisão Permissão-Papel**
Flat	Sim	Não
Flat	Não	Sim
Flat	Sim	Sim
Hierárquico Geral	Não	Sim
Hierárquico Limitado	Não	Sim

3.3.2. RBAC Básico

Os aspectos essenciais e fundamentais do RBAC estão definidos no RBAC Básico. O conceito básico do RBAC é a associação de usuários a papéis, a associação de permissões a papéis e a aquisição de permissões do usuário pelo papel que o mesmo desempenha. Além do conceito básico do RBAC o modelo RBAC Básico ainda exige que as associações usuário-papel e permissão-papel possam ser muitos-para-muitos, suporte à revisão usuário-papel, e a ativação múltipla de papéis.

O RBAC Básico mostrado na Figura 3.1 possui três conjuntos de entidades: usuários (U), papéis (R, de *roles*) e permissões (P). Neste modelo um usuário é uma pessoa ou um

processo agindo em nome de uma pessoa. Um papel é uma função ou cargo dentro da organização que possui uma semântica representando a autoridade e a responsabilidade conferidas aos membros desse papel. Uma permissão é um direito específico de acesso a um ou mais objetos do sistema; a natureza exata de uma permissão é dependente da implementação e não é especificada pelo modelo RBAC-NIST [SAN00].

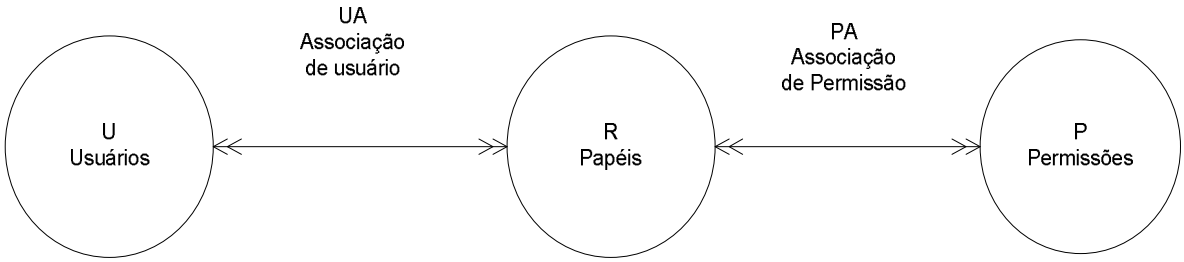


Figura 3.1: RBAC Básico

O RBAC Básico exige que a associação usuário-papel (UA, de *user-role assignment*) e a associação permissão-papel (PA, de *permission-role assignment*) sejam relações do tipo muitos-para-muitos (isto é indicado pelas setas duplas na Figura 3.1). Este é um aspecto essencial do RBAC pois permite que um usuário exerça as permissões de vários papéis. O princípio do mínimo privilégio dificilmente pode ser respeitado quando todos os papéis de um usuário são ativados em uma sessão, mas esse princípio pode ser respeitado mais facilmente permitindo-se o usuário ativar e desativar os papéis que deseja utilizar, pois o usuário pode ativar apenas os papéis necessários à execução de determinada tarefa.

O suporte à revisão usuário-papel possibilita determinar de maneira eficiente quais os usuários associados a um papel e a quais papéis um usuário está associado. A revisão permissão-papel também é muito importante no RBAC para responder quais permissões estão associadas a um papel e a quais papéis uma permissão está associada, mas pela dificuldade de se implementar este mecanismo em sistemas distribuídos de larga escala, ele é requerido somente no RBAC Simétrico [SAN00].

3.3.3. RBAC Hierárquico

A inclusão da relação de hierarquia de papéis (RH, de *role hierarchy*), mostrada na Figura 3.2 é a principal diferença do RBAC Hierárquico para o Básico. Uma hierarquia é

matematicamente uma ordem parcial definindo uma relação de precedência de papéis, por meio da qual papéis de categoria superior adquirem as permissões dos seus subordinados. Hierarquias de papéis são uma forma natural de estruturar os papéis de modo a refletir as linhas de autoridade e responsabilidade em uma organização. O modelo do NIST divide a RBAC Hierárquico em dois níveis:

- RBAC Hierárquico Geral: neste caso qualquer tipo de ordem parcial pode constituir uma hierarquia de papéis.
- RBAC Hierárquico Limitado: neste caso existem restrições em relação à estrutura da hierarquia de papéis. Geralmente, as hierarquias são limitadas a estruturas simples como árvores ou árvores invertidas.

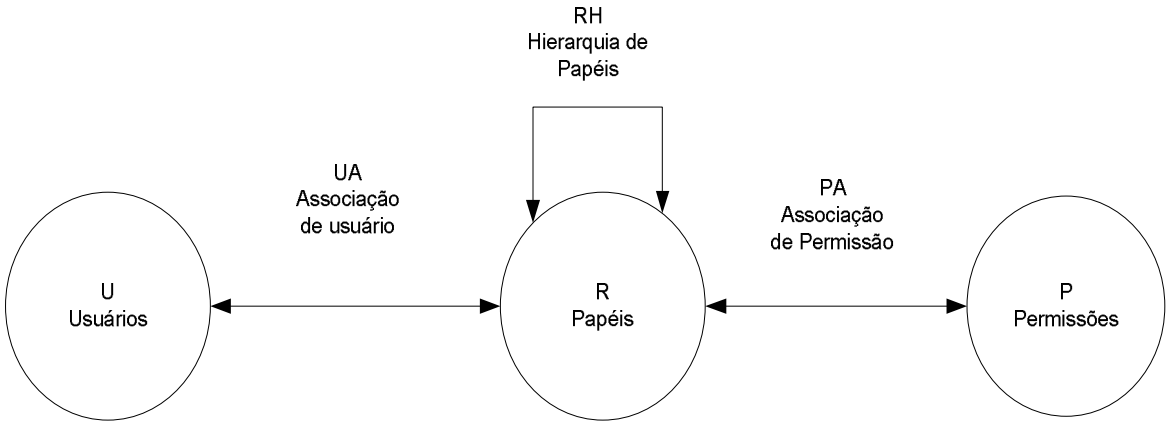


Figura 3.2: RBAC Hierárquico

O mecanismo de revisão usuário-papel do RBAC Básico deve ser estendido para suportar hierarquias de papéis permitindo a identificação tanto dos papéis associados diretamente a um usuário (definidos pelo administrador de segurança) como dos papéis associados indiretamente ao usuário (cuja associação se dá através de herança).

As Figuras a seguir mostram um exemplo de hierarquia em árvore invertida (Figura 3.3) , hierarquia em árvore (Figura 3.4) e hierarquia geral (Figura 3.5) de um departamento de engenharia fictício

A Figura 3.3 mostra um exemplo de hierarquia em árvore invertida. Aqui, Engenheiro de Produção 1 e Engenheiro de Qualidade 1 compartilham permissões de Engenheiro 1, que, por sua vez, compartilha com Engenheiro 2 as permissões do Departamento de Engenharia.

Como pode ser visto, árvores invertidas são boas para o compartilhamento de permissões através de papéis, mas não permitem a agregação de permissões de vários papéis.

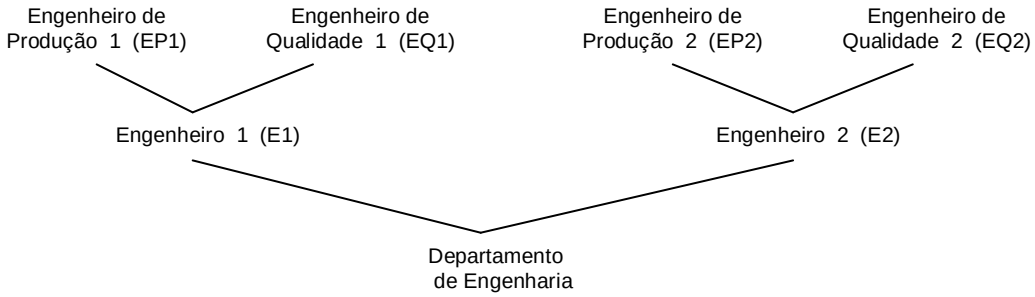


Figura 3.3: Exemplo de RBAC Hierárquico - Árvore Hierárquica Invertida

A Figura 3.4 mostra um exemplo de hierarquia em árvore. Nesta hierarquia, os papéis mais privilegiados (isto é, com mais permissões) ocupam posições mais altas; estes papéis herdam as permissões dos papéis localizados abaixo deles na árvore. No exemplo, o Diretor tem, além de suas próprias permissões, as permissões herdadas de Líder de Projeto 1 e Líder de Projeto 2. O Líder de Projeto 1 tem as suas permissões e mais as de Engenheiro de Produção 1 e Engenheiro de Qualidade 1. Este tipo de hierarquia permite a agregação de permissões de diferentes papéis em um outro papel; entretanto, ela não possibilita o compartilhamento de permissões através de um papel.

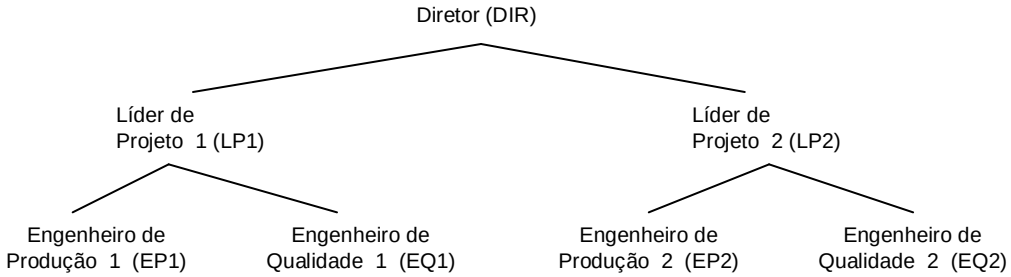


Figura 3.4: Exemplo de RBAC Hierárquico - Árvore Hierárquica

A Figura 3.5 mostra um exemplo de hierarquia geral. Neste caso, tanto a agregação quanto o compartilhamento de permissões são possíveis. Cabe salientar que na prática as

hierarquias de papéis tendem a ser bem mais irregulares do que a hierarquia simétrica da Figura 3.5.

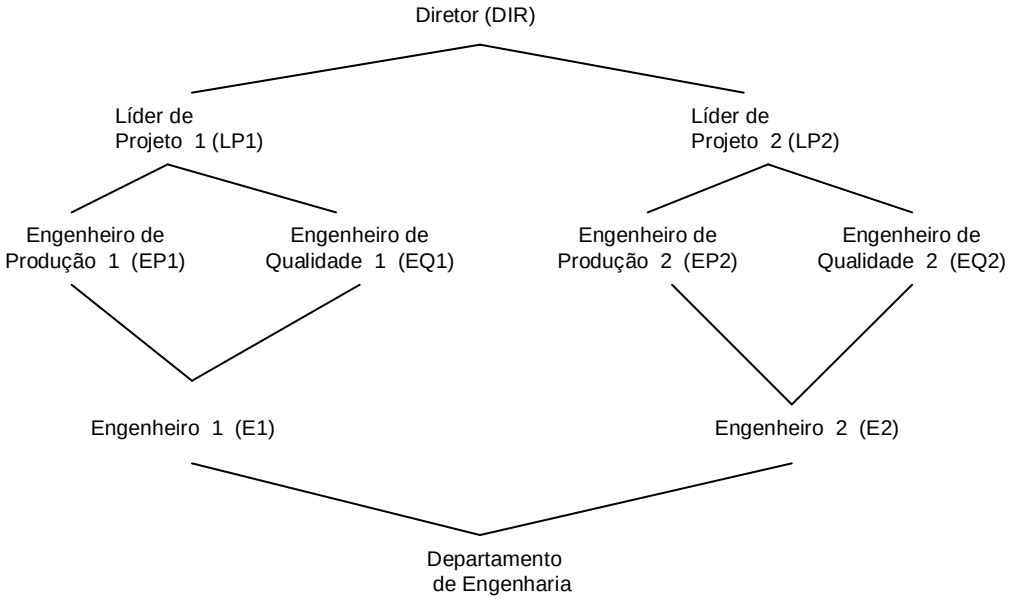
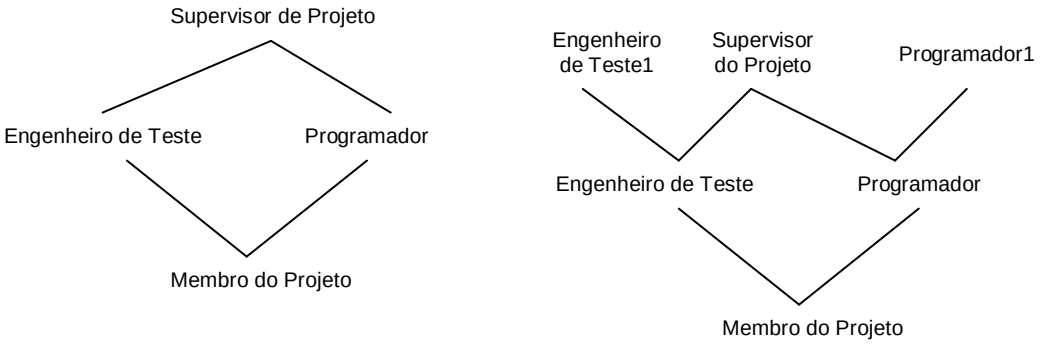


Figura 3.5: Exemplo de RBAC Hierárquico - Hierarquia Geral

Papéis no topo da hierarquia, como Diretor na Figura 3.5, são considerados perigosos pelo fato de concentrarem muito poder [SAN96]. Mesmo que os usuários associados a esses papéis sejam bastante confiáveis, os danos causados por um erro que eles venham a cometer ou pela ação de um software malicioso que eles venham a utilizar são, potencialmente, muito grandes. É possível limitar a herança de permissões em hierarquias de papéis; a Figura 3.6 mostra como isso pode ser feito.

Na figura 3.6(a), Supervisor de Projeto herda todas as permissões dos papéis inferiores. Por outro lado, na Figura 3.6(b) os Engenheiros de Teste podem ter permissões no papel Engenheiros de Teste1 que não são herdadas por Supervisor do Projeto. Os usuários são, na verdade, associados ao papel Engenheiros de Teste1; o papel Engenheiro de Testes serve somente para conter as permissões que devem ser compartilhadas. Papéis como Engenheiros de Teste1 são chamados papéis privados. O papel Programador1 também é um papel privado, e a discussão acima aplica-se igualmente ao mesmo.



(a) (b)

Figura 3.6: Exemplo de Herança Limitada

Existem duas interpretações distintas para uma hierarquia de papéis discutidas na literatura. Na primeira delas, papéis superiores (*senior roles*) herdam as permissões dos papéis inferiores (*junior roles*); neste caso, tem-se uma hierarquia de herança. Se a Figura 3.5 for vista como uma hierarquia de herança, quando o papel Líder de Projeto 1 é ativado ele herda todas as permissões de Engenheiro de Produção 1, Engenheiro de Qualidade 1, Engenheiro 1 e Departamento de Engenharia.

Na outra interpretação para a hierarquia de papéis, a ativação de um papel superior não implica na ativação automática das permissões dos papéis inferiores; neste caso, tem-se uma hierarquia de ativação. Para que as permissões dos papéis inferiores sejam ativadas, estes papéis devem ser explicitamente ativados.

É possível aplicar as duas interpretações simultaneamente. Em tais casos, a hierarquia de ativação pode estender a hierarquia de herança ou ser independente desta [SAN98b]. O modelo RBAC-NIST não define uma interpretação específica para as hierarquias de papéis [SAN00].

3.3.4. RBAC com Restrições

Um conceito bastante importante para o modelo de restrições RBAC, no sentido de minimizar a ocorrência de erros e fraudes na manipulação da informação, é a separação de tarefas (*separation of duty*). Este conceito consiste em dividir tarefas que pode gerar conflito de interesses em várias subtarefas menores, executadas por pessoas diferentes, reduzindo, desta maneira, o poder individual de cada usuário. A separação de tarefas teve a sua importância para a segurança da informação reconhecida e discutida em detalhes por Clark e

Wilson [CLA87]. O RBAC facilita a implantação de separação de tarefas, utilizando-se, para isso, de relações de exclusão mútua entre papéis.

Existem duas formas de separação de tarefas, estática e dinâmica. Na separação estática de tarefas, dois papéis R1 e R2 que são mutuamente exclusivos não podem ter usuários em comum; em outras palavras, um mesmo usuário não pode ser associado a R1 e a R2. Por outro lado, na separação dinâmica de tarefas uma exclusão mútua entre dois papéis R1 e R2 significa que um usuário pode ser associado a ambos, desde que apenas um deles (R1 ou R2) esteja ativo em um dado momento [SAN00].

Este conceito será estudado mais detalhadamente no capítulo seguinte.

3.3.5. RBAC Simétrico

Um componente essencial para qualquer esquema de gerenciamento de autorização é a manutenção apropriada e precisa das associações permissão-papel. O RBAC Simétrico possui como único requisito o suporte à revisão de associações permissão-papel similar à revisão usuário-papel do RBAC Básico. O desempenho de revisões permissão-papel deve ser comparável ao desempenho de revisões usuário-papel.

A revisão permissão-papel possibilita identificar, de maneira eficiente, as permissões associadas a um papel e, também, quais papéis possuem determinadas permissões. A possibilidade de identificar as associações permissão-papel é um dos principais fatores que diferenciam papéis de grupos. Apesar da sua importância, a revisão permissão-papel só é exigida no último nível pela dificuldade e complexidade de sua implementação em ambientes distribuídos de larga escala.

3.3.6. Outras características do modelo RBAC-NIST

O modelo unificado RBAC-NIST não abrange ou contempla parcialmente algumas características do RBAC justificando a não abordagem de tais características por três razões. A primeira, porque não julga apropriado especificar em um modelo detalhes dependentes da aplicação tais como: escalabilidade, permissões negativas, natureza das permissões (granularidade), ativação seletiva de papéis, revogação de papéis. E a segunda porque não existe consenso suficiente da comunidade para justificar a inclusão de algumas características no modelo como, por exemplo, administração e definição de outras restrições além da separação de tarefas.

O modelo RBAC-NIST reconhece, no nível 3, apenas restrições de separação de tarefas (tanto estática como dinâmica). O RBAC pode suportar outros tipos de restrições, como pré-condições (determinam as condições que devem ser satisfeitas para que as associações usuário-papel e papel-permissão possam ser efetuadas [SAN98b]), restrições de cardinalidade (determinados papéis podem ter um número máximo de usuários associados [FER99]) e obrigações (atributos funcionais verificam as exigências que um sujeito deve atender antes ou durante um acesso [AHN99, SAN03]). Embora estes tipos de restrições sejam igualmente importantes, as restrições de cardinalidade e pré-condições não foram incluídas no modelo unificado RBAC-NIST por não existir ainda consenso sobre a sua real utilização e as restrições de obrigações foram consideradas muito recentes. As restrições serão estudas mais detalhadamente no próximo capítulo.

3.3.7. Especificação Funcional do Modelo Unificado RBAC-NIST

O modelo unificado RBAC-NIST define uma série de entidades, componentes e relacionamentos. A especificação funcional define as funções necessárias para a criação e manutenção dos componentes do modelo RBAC. A proposta de especificação de [FER01] é composta por três categorias de funções:

- Funções Administrativas – Funções de criação e manutenção dos elementos e relacionamentos para a construção dos vários modelos RBAC;
- Funções de Suporte ao Sistema – Funções de suporte necessárias para o gerenciamento das sessões e para as decisões de controle de acesso;
- Funções de Revisão – Funções de verificação dos resultados das ações realizadas pelas funções administrativas;

Especificação Funcional para o RBAC Básico

As funções administrativas têm por objetivo criar e manter os elementos básicos do RBAC Básico. Desta forma, administradores do sistema podem criar e excluir usuários e papéis, e estabelecer relacionamentos entre papéis e operações ou objetos. As funções administrativas para os usuários são: *AddUser* e *DeleteUser*. Para os papéis são: *AddRole* e *DeleteRole*.

As funções administrativas para a criação e manutenção de relacionamentos entre usuário e papel são: *AssignUser* e *DeassignUser*. Para o relacionamento entre as permissões e os papéis são: *GrantPermission* e *RevokePermission*.

As Funções de Suporte ao Sistema são necessárias para o gerenciamento das sessões e para as decisões de controle de acesso. Para que um usuário passe pelo processo de autenticação e entre na sessão, é necessário que haja pelo menos um papel ativo para o mesmo. É possível, durante a sessão, que o usuário altere a disponibilização de seus papéis a serem utilizados, os quais foram previamente definidos. Estas funções são:

- *CreateSession*: cria a sessão do usuário e disponibiliza ao usuário seus papéis ativos;
- *AddActiveRole*: adiciona o papel, previamente definido, como ativo na sessão atual;
- *DropActiveRole*: exclui o papel ativo, previamente definido, da atual sessão;
- *CheckAccess*: determina se o usuário tem permissão para executar uma determinada operação ou acessar um objeto.

As funções de revisão permitem a visualização das relações entre usuários e papéis, e permissões e papéis previamente definidos. Desta forma, deve ser possível, por exemplo, que um administrador do sistema visualize todos os usuários que exercem um determinado papel ou então todos os papéis que um determinado usuário exerce. As funções de revisão são dividas em funções que têm sua implementação obrigatória (*mandatory)* e funções com implementação opcional. As funções de revisão obrigatórias (M) e opcionais (O) são:

- *AssignedUsers* (M): retorna um conjunto de usuários associados a um determinado papel;
- *AssignedRoles* (M): retorna um conjunto de papéis associados com um determinado usuário;
- *RolesPermissions* (O): retorna um conjunto de permissões atribuídas a um determinado papel;
- *UserPermissions* (O): retorna um conjunto de permissões que um usuário pode executar através de seus papéis.
- *SessionRoles* (O): retorna o conjunto de papéis ativos da sessão.
- *SessionPermissions* (O): retorna o conjunto de permissões disponíveis na sessão.

- *RoleOperationsonObject* (O): retorna o conjunto de operações que um determinado papel pode realizar em um determinado objeto.
- *UserOperationsOnObject* (O): retorna o conjunto de operações que um determinado usuário pode realizar em um determinado objeto.

Especificação Funcional para o RBAC Hierárquico

O RBAC Hierárquico requer, além das funções administrativas do RBAC Básico, funções administrativas para manter a ordem parcial nas relações entre papéis. As novas funções administrativas são:

- *AddInheritance*: é responsável pela criação de um relacionamento hierárquico entre dois papéis existentes.
- *DeleteInheritance*: exclui um relacionamento hierárquico entre dois papéis existentes.
- *AddAscendant*: cria um novo papel e o insere como ascendente de um papel já existente.
- *AddDescendant*: cria um novo papel e o insere como descendente de um papel já existente.

As Funções de Suporte ao Sistema do RBAC Hierárquico são as mesmas do RBAC Básico. Entretanto, por causa da hierarquia de papéis, as funções *CreateSession* e *AddActiveRole* são redefinidas. O conjunto de papéis ativados pela função *CreateSession* deve conter, além dos papéis associados diretamente ao usuário, alguns ou todos os papéis herdados pelos papéis diretamente associados ao usuário. Do mesmo modo, na função *AddActiveRole*, um usuário pode ativar um papel associado diretamente ou papéis herdados pelos papéis associados diretamente.

Todas as funções de revisão do RBAC Básico são válidas para o RBAC Hierárquico. São necessárias mais duas funções:

- *AuthorizedUsers:* retorna o conjunto de usuários associados diretamente a um determinado papel e o conjunto de usuários associados aos papéis herdados do papel determinado.
- *AuthorizedRoles*: retorna o conjunto de papéis diretamente associados a um determinado usuário e o conjunto de papéis herdados dos papéis associados diretamente ao usuário.

Especificação funcional para Separação Estática de Tarefa

As funções administrativas para o Modelo RBAC com Restrições SSD (*Static Separation of Duty* – Separação Estática de Tarefas*)* incluem todas as funções do modelo RBAC Básico e funções para criação e manutenção de relações de SSD. A função administrativa *AssignUser* do RBAC Básico deve incorporar a funcionalidade para verificar se a associação de um determinado usuário a um determinado papel não viola as restrições de SSD. As funções administrativas para manutenção e criação de relações de SSD são:

- *CreateSSDSet*: cria um novo conjuntos de relacionamentos SSD.
- *DeleteSSDSet*: exclui um conjuntos de relacionamentos SSD.
- *AddSSDRoleMember*: insere um papel como membro de um conjunto SSD existente.
- *DeleteSSDRoleMember*: exclui um papel de um conjunto SSD existente.
- *SetSSDCardinality*: especifica a cardinalidade da composição dos papéis do conjunto SSD correspondente.

Todas as funções de revisão do modelo RBAC Básico são necessárias para implementação do modelo RBAC com Restrições SSD. Além disso, funções para visualização do resultado das funções administrativas adicionais são necessárias:

- *SSDRoleSets*: retorna o conjunto de relações SSD existentes.
- *SSDRoleSetRoles*: retorna o conjunto de papéis associados a um conjunto SSD.
- *SSDRoleSetCardinalitty*: retorna a cardinalidade do conjunto SSD correspondente.

Especificação funcional para Separação Dinâmica de Tarefa

A semântica para criação de instâncias de relações DSD (*Dynamic Separation of Duty* – Separação Dinâmica de Tarefas) é idêntica a de relações SSD. Enquanto restrições SSD são aplicadas durante a associação de usuários, as restrições DSD são aplicadas somente no momento da ativação do papel na sessão do usuário. As funções administrativas adicionais necessárias no Modelo RBAC com Restrições DSD são:

- *CreateDSDSet*: cria um novo conjuntos de relacionamentos DSD.
- *DeleteDSDSet*: exclui um conjuntos de relacionamentos DSD.
- *AddDSDRoleMember*: insere um papel como membro de um conjunto DSD existente.

- *DeleteDSDRoleMember*: exclui um papel de um conjunto DSD existente.
- *SetDSDCardinality*: especifica a cardinalidade da composição dos papéis do conjunto DSD correspondente.

As funções de suporte ao sistema *CreateSession* e *AddActiveRole* devem incorporar a funcionalidade de verificar restrições DSD.

Assim como no modelo RBAC com Restrições SSD, todas as funções de revisão do modelo RBAC Básico são necessárias para implementação do modelo RBAC com Restrições DSD. Além disso, funções para visualização do resultado das funções administrativas adicionais são necessárias:

- *DSDRoleSets*: retorna o conjunto de relações DSD existentes.
- *DSDRoleSetRoles*: retorna o conjunto de papéis associados a um conjunto DSD.
- *DSDRoleSetCardinalitty*: retorna a cardinalidade do conjunto DSD correspondente.

3.4. Outros Modelos RBAC

Esta seção mostra alguns dos principais modelos RBAC publicados na literatura, que apesar de compartilharem um núcleo comum de conceitos, possui particularidades que os diferenciam.

3.4.1. O Modelo do NIST

O primeiro modelo RBAC formalizado na literatura foi o de Ferraiolo e Kuhn [FER92], do NIST, que passou por revisões, e atualizações de alguns de seus conceitos ao longo do tempo [FER95, FER99].

O modelo do NIST tem como principais características possuir suporte a restrições de cardinalidade e ser um modelo único, e não uma família como o modelo unificado, bastante similar ao RBAC com Restrições da família RBAC-NIST.

3.4.2. A Família RBAC96

O modelo proposto por Sandhu et all. [SAN96] foi o primeiro a reconhecer a impossibilidade de capturar todas as variações do RBAC em um único modelo, o que levou à definição da família RBAC96 de modelos. Esta família, como no modelo unificado RBAC-NIST, possui quatro modelos. Os modelos são praticamente iguais aos do modelo

unificado na interpretação não ordenada. A principal diferença entre o RBAC Básico e o RBACo é a introdução do conceito de sessões no modelo RBACo [SAN96].

Os modelos RBAC 2 e RBAC3 (equivalentes aos modelos RBAC com Restrições e RBAC Simétrico do modelo unificado) suportam, além das restrições de separação de tarefas, restrições de cardinalidade e de pré-condições (um usuário U só pode ser associado a um papel R1 se já estiver associado também a um papel R2, ou uma permissão P1 só pode ser associada a um papel R, se uma outra permissão P2 já estiver também associada ao papel R). A família RBAC96 possui, ainda, um modelo administrativo associado denominado ARBAC97 [SAN98b, SAN99].

3.5. Implementações de RBAC

As implementações práticas seguiram-se aos modelos propostos. Houve proposições de implementações para várias situações a saber:

- Proposição de controle de acesso a bancos de dados cliente-servidor Oracle 7 através do modelo RBAC em nível de transações [ORA92]
- Proposição de controle de acesso aos objetos de uma Intranet corporativa (RBAC/Web) [BAR97] [FER99]
- Proposição de modelo de 3 validações, onde o acesso a um objeto seria regulado inicialmente pela política RBAC. Em caso de autorização solicitar-se-ia o acesso ao mesmo objeto através da política DAC (*Discricionary Access Control*) e, em seguida à MAC (*Mandatory Access Control*) [FRI97].
- Proposição de uma extensão do JAAS (*Java Authentication and Authorization Service*), modelo de segurança Java, para permitir a ativação de papéis sem intervenção do usuário (JRBAC-Web) [GIU99]
- Proposição de um modelo que pode ser implementado em sistemas de objetos distribuídos que seguem os padrões OMG/CORBA. O modelo proposto permite a ativação automática de papéis pelos componentes de segurança do *middleware* trazendo o controle de acesso baseado em papéis para aplicações não cientes da segurança (RBAC-JaCoWeb) [OBE01].

3.6. Conclusões do Capítulo

Este capítulo apresentou o modelo de controle de acesso baseado em papéis que permite simplificar a administração da política de segurança facilitando a definição de políticas flexíveis e centralizadas. O RBAC agregou as melhores características dos modelos clássicos (discricionário e obrigatório) no sentido de ser flexível como o primeiro e centralizado como o segundo, o que é difícil de se conseguir num mesmo modelo.

Apesar de ser um conceito da década de 70, somente em 1992 o RBAC começou a ser formalizado. A partir daí surgiram variações de modelos RBAC. O primeiro passo na direção de uma padronização foi dado pelo NIST (*National Institute of Standards and Technology*) com a proposta do modelo unificado RBAC-NIST apresentado na seção 3.3.

Por ser uma tecnologia que só recentemente vem sendo explorada o RBAC necessita de estudos mais aprofundados em algumas áreas. Uma área carente de pesquisas é o modelo de restrições que aborda somente restrições de separação de tarefas. Outros tipos de restrições igualmente importantes, como as restrições de cardinalidade e pré-condições não foram incluídas no modelo unificado RBAC-NIST por não existir ainda consenso sobre a sua real utilização e as restrições de obrigações foram consideradas muito recentes.

Capítulo 4

Modelos de Restrições do RBAC

As restrições são um aspecto importante nos modelos de controle de acesso baseado em papéis e freqüentemente são consideradas como uma das principais motivações do RBAC. Embora a importância das restrições do RBAC tenha sido reconhecida há muito tempo, até o momento não existem muitos trabalhos no sentido de expandir o conceito de restrições. A maior parte dos modelos propostos abordam somente restrições de separação de tarefas. Outros tipos de restrições raramente são abordados.

Este capítulo apresenta o modelo de restrições do Modelo Unificado RBAC-NIST, o modelo de restrições do Modelo de Controle de Uso ABC e outras propostas de modelos de restrições.

4.1. Modelo de Restrições do Modelo Unificado RBAC-NIST

O RBAC com restrições, mostrado nas Figuras 4.1 e 4.2, acrescenta restrições ao modelo RBAC Hierárquico. O modelo RBAC-NIST formaliza apenas as restrições do princípio de separação de tarefas (SOD, de *separation of duty*) por ser uma das características mais desejadas de um esquema de controle de acesso [FER92b]. A separação de tarefas é suportada, de alguma forma, por vários modelos de RBAC [NYA99,SAN96]. Na família RBAC-NIST, a separação de tarefas é suportada pelo modelo RBAC com Restrições.

A principal motivação da separação de tarefas é minimizar a ocorrência de erros e fraudes na manipulação da informação. Este conceito consiste em dividir tarefas que podem gerar conflitos de interesses [BRE89] em várias subtarefas menores, executadas por pessoas diferentes, reduzindo, desta maneira, o poder individual de cada usuário. A importância da separação de tarefas para a segurança da informação foi reconhecida e discutida em detalhes

por Clark e Wilson [CLA87]. A separação de tarefas é suportada pelo princípio do mínimo privilégio na definição dos papéis, ou seja, os papéis têm que estar associados ao mínimo de permissões necessárias ao cumprimento de suas tarefas.

4.1.1. Separação Estática de Tarefas

O conflito de interesses em um sistema baseado em papéis pode surgir como resultado da obtenção de permissões associadas com papéis conflitantes [SAN00]. Uma forma de prevenir esta forma de conflito de interesse é através da separação estática de tarefas (SSD – *Static Separation of Duty*), ou seja, impondo restrições à associação de usuários a papéis (Figura 4.1). Nesta abordagem, usuários associados a um papel não podem ser associados a um segundo papel, de acordo com restrições definidas pelo administrador de segurança. Por exemplo, o papel Compras (associado aos funcionários do setor de compras) pode ser definido como mutuamente exclusivo em relação ao papel Almoxarifado (Figura 4.3). Esta restrição impede que um funcionário desonesto use o papel Compras para forjar uma requisição de compra de um produto e, a seguir, usando o papel Almoxarifado, faça um registro fraudulento de entrada do produto no estoque do almoxarifado, o que lhe possibilitaria embolsar o valor referente à compra.

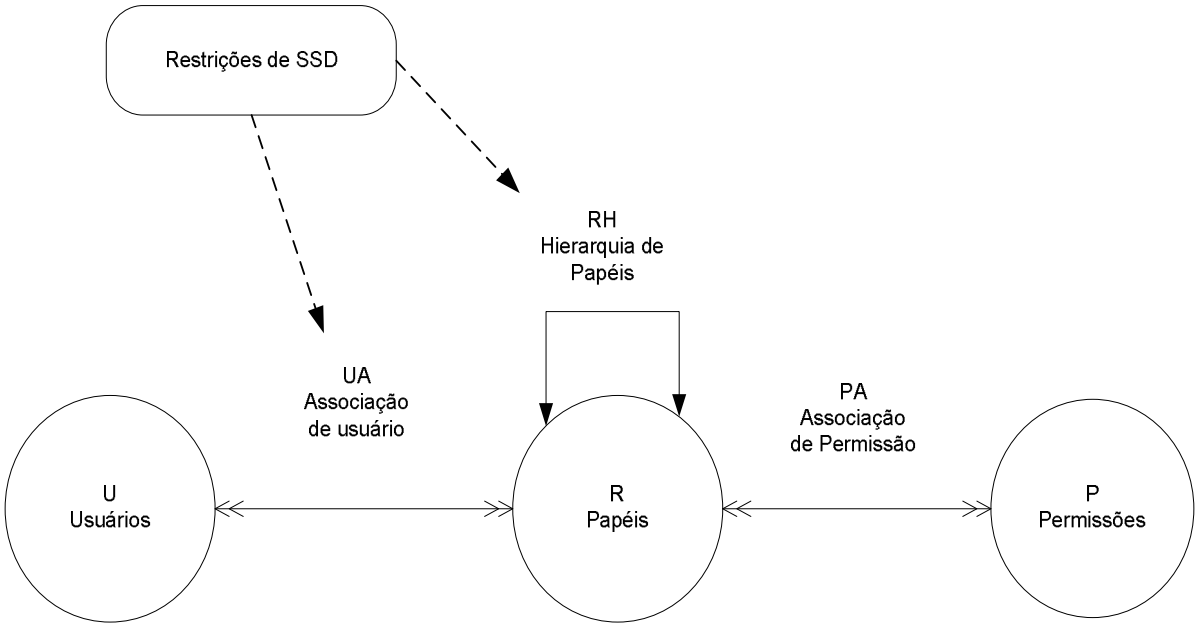


Figura 4.1: RBAC com Restrições – Separação Estática de Tarefas

Nota-se que estas restrições se propagam através de uma hierarquia de papéis. Por exemplo, se o papel Supervisor de Compras herda o papel Compras, então ele também é mutuamente exclusivo em relação a Almoxarifado; o mesmo vale para Compras em relação a Chefe de Almoxarifado e também para este em relação a Supervisor de Compras. Como um papel superior é, na verdade, uma instância de um papel inferior, não pode existir SOD estática entre eles. No exemplo da Figura 4.3, não faz sentido estabelecer uma exclusão entre Supervisor de Compras e Compras, já que, por definição, não pode haver conflito de interesse entre eles. Caso houvesse, a relação de herança não deveria ter sido usada [FER99, SAN00].

4.1.2. Separação Dinâmica de Tarefas

O uso de políticas de separação dinâmica de tarefas (DSD – *Dynamic Separation of Duty*) (Figura 4.2) também é permitido no RBAC com Restrições. Nesta abordagem, os usuários podem ser associados a papéis que só constituem um conflito de interesse quando ativados simultaneamente, ou seja, é perfeitamente possível e seguro que um usuário ative mais de um papel do conjunto, desde que esta ativação não seja simultânea.

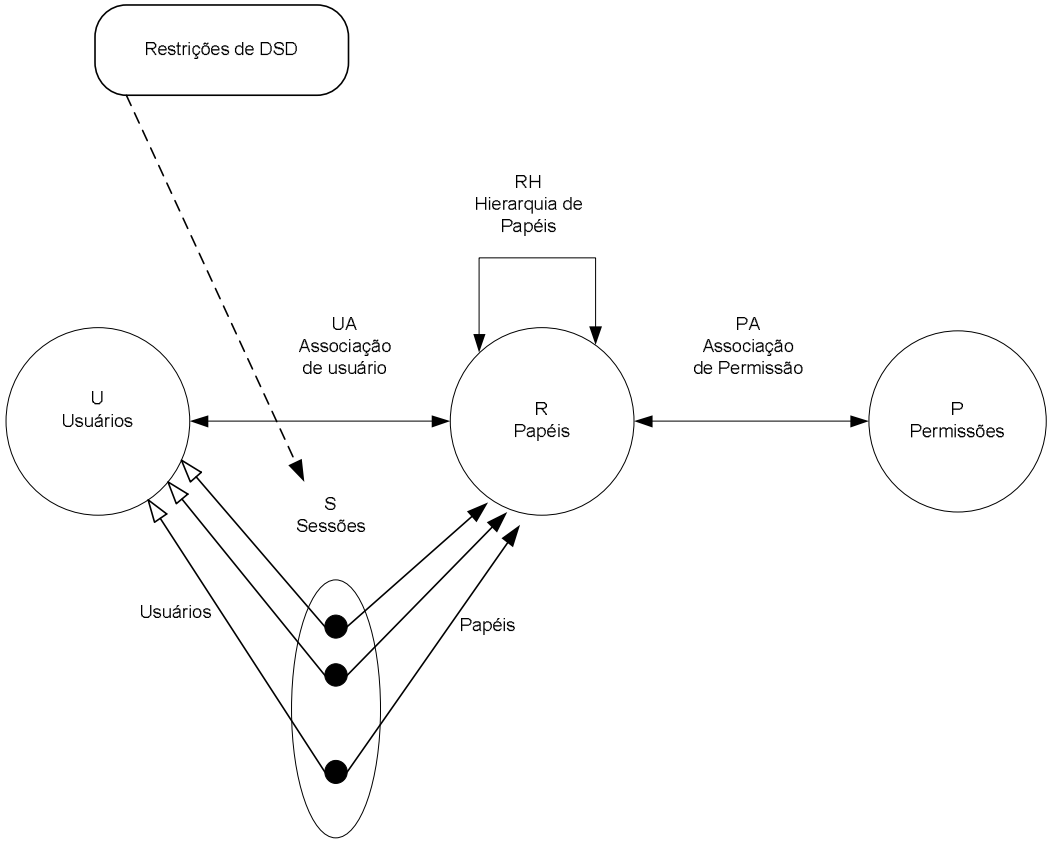


Figura 4.2: RBAC com Restrições – Separação Dinâmica de Tarefas

No exemplo da Figura 4.3, um mesmo usuário pode ser associado aos papéis Contador e Contador-Chefe, onde o Contador lança os registros contábeis e o Contador-Chefe pode efetuar correções em lotes de lançamentos ainda não consolidados. Se um usuário agindo como Contador tentasse ativar o papel Contador-Chefe, ele teria que desativar primeiro o papel Contador — forçando a consolidação dos lançamentos — antes que o indivíduo assumisse o papel Contador-Chefe. Desde que o mesmo usuário não possa ativar ambos papéis ao mesmo tempo, não há conflito de interesse. Embora seja possível obter a exclusão através de uma separação estática de tarefas, a separação dinâmica é, geralmente, mais flexível.

A separação dinâmica de tarefas pode ser perfeitamente aplicada a papéis que tenham uma relação de herança, ao contrário do que acontece na separação estática [SAN00].

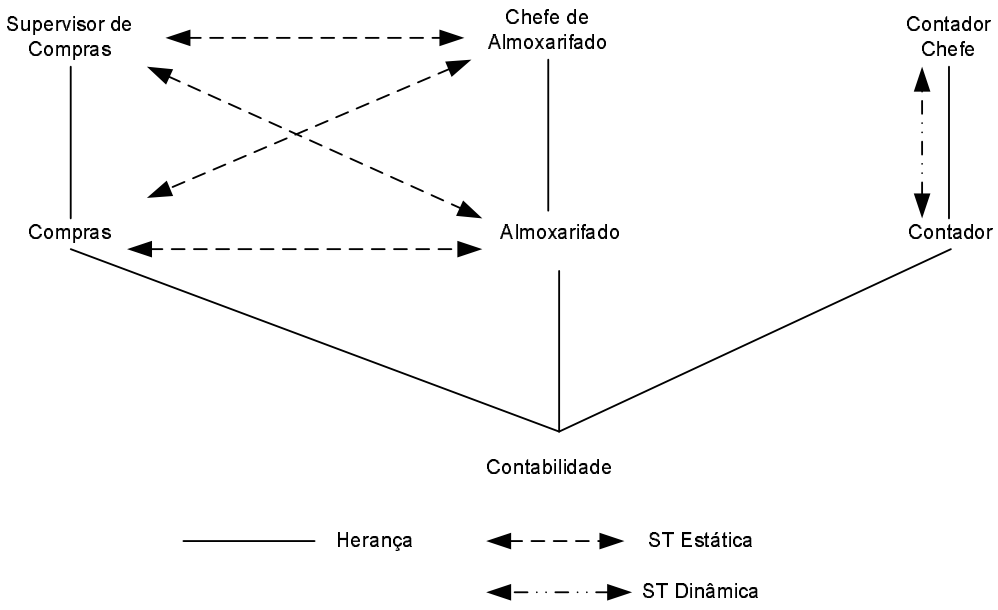


Figura 4.3: RBAC com Restrições – Exemplo de Separação de Tarefas

Dinâmica do estabelecimento de uma sessão RBAC com Separação Dinâmica de Tarefas

Na Figura 4.4 é mostrada uma visão geral da dinâmica do estabelecimento de uma sessão RBAC, onde são aplicadas restrições de DSD. No diagrama de colaboração do processo de ativação da sessão, o usuário solicita a inicialização da sessão (evento S1),

acionando o controlador RBAC através da interface de usuário. No evento S1.2 são recuperados os papéis associados ao usuário e em S1.4 são recuperados os sub-papéis (papéis derivados da hierarquia, se for o caso) que o usuário pode ativar na sessão (evento S1.7). As restrições de ativação são recuperadas no evento S2.2, então o controlador RBAC verifica se o usuário possui autorização para ativar os papéis que escolheu no evento S2, considerando as restrições de DSD. Se todos os passos comentado anteriormente ocorrerem com sucesso, os papéis selecionados são ativados e a sessão é aberta para o usuário (evento S2.7).

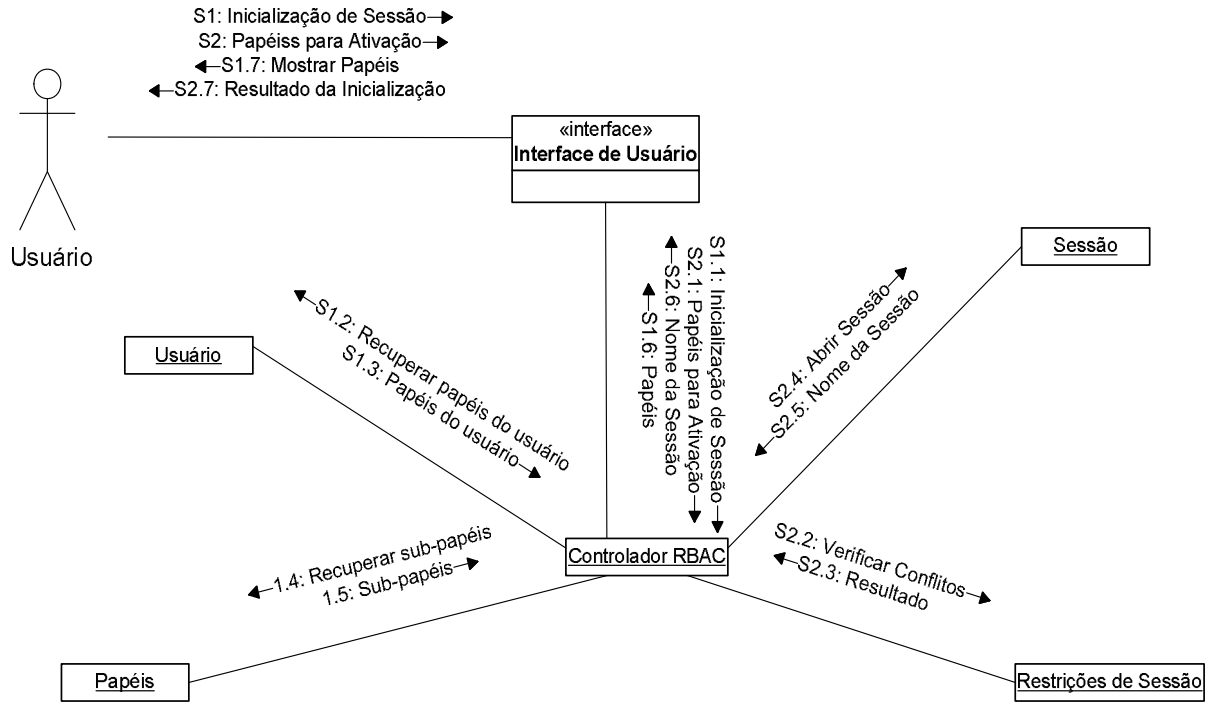


Figura 4.4: Diagrama de colaboração - Ativação de Sessão

4.1.3. Diagrama de classes do Modelo Unificado RBAC-NIST

O modelo estático conceitual, representado pelo diagrama de classes da Figura 4.5, possibilita um melhor entendimento do Modelo Unificado RBAC-NIST com Restrições. Ele contém classes, relações entre classes e cardinalidade nos relacionamentos. Os papéis e as permissões são especializados em duas categorias: usuários e administrativos. Os principais com papéis administrativos possuem permissões administrativas, ou seja, podem associar principais com papéis (UA) e papéis com permissões (PA). Os principais associados a papéis

de usuário possuem permissões de usuário e não podem realizar operações administrativas. As restrições são divididas em SSD e DSD. As restrições de SSD são aplicadas aos papéis administrativos no momento da associação de usuários a papéis, enquanto as restrições de DSD são aplicadas na inicialização de uma sessão e na inclusão de um papel em uma sessão inicializada.

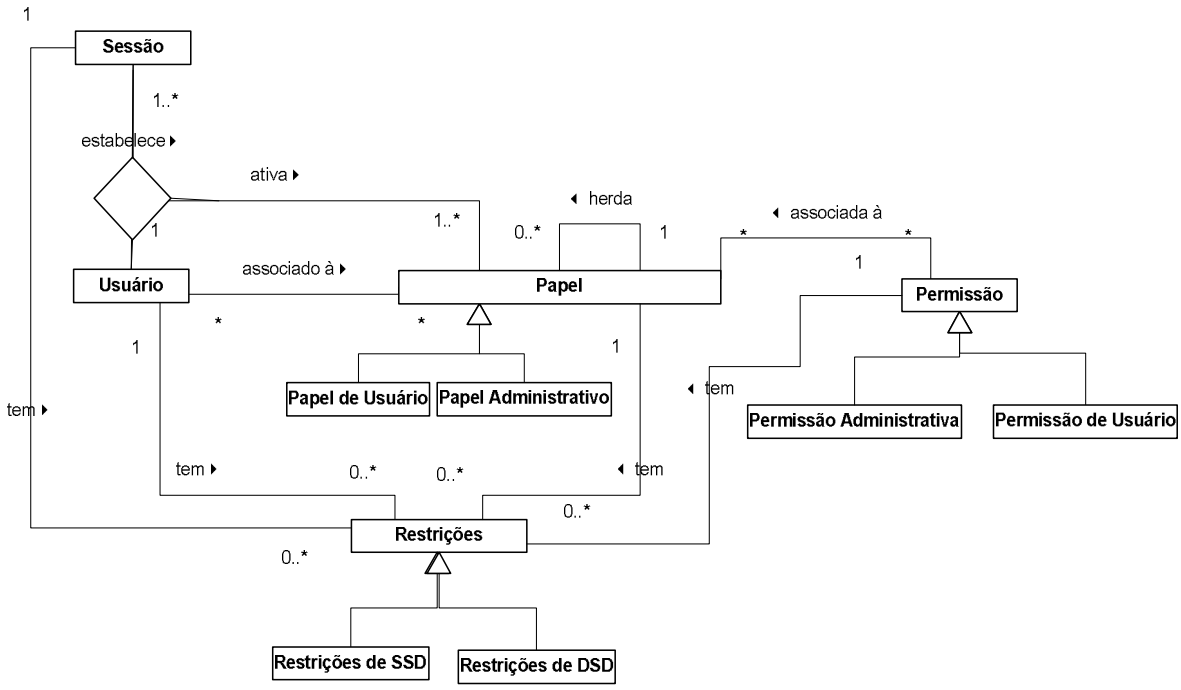


Figura 4.5: Diagrama de Classes – RBAC com Restrições

4.2. Modelo de Restrições do Modelo $UCON_{ABC}$

A família de modelos para controle de uso ($UCON_{ABC}$) denominada ABC (*Authorizations, oBligations, and Conditions*) é apresentada por [PAR02, SAN03, PAR04]. O termo controle de uso é uma generalização de controle de acesso para cobrir obrigações, condições, continuidade (controles contínuos) e mutabilidade. Tradicionalmente, o controle de acesso é realizado baseado somente nas autorizações, ou seja, nos atributos do sujeito, nos atributos do objeto que o sujeito quer acessar e na operação que o sujeito deseja realizar no objeto. O modelo ABC baseia-se em três fatores de decisão: autorização, obrigações e condições. Obrigações são exigências que devem ser cumpridas pelos sujeitos antes ou durante o acesso. Condições são exigências do ambiente de sujeitos e objetos (independentes)

que devem ser satisfeitas para acesso. As obrigações podem ser aplicadas antes do acesso ou durante o acesso, enquanto as condições só podem ser aplicadas antes do acesso.

O modelo $UCON_{ABC}$ é composto por oito componentes principais (Figura 4.6): sujeitos (*subject*), atributos de sujeitos, objetos, atributos de objetos, direitos, autorizações, obrigações e condições. Autorizações, obrigações e condições são atributos funcionais que devem ser avaliados na decisão de uso. Controles de acesso tradicionais utilizam só autorizações no processo de decisão. Obrigações e condições são conceitos novos que estão sendo discutidos para solucionar algumas falhas recentemente mostradas em controles de acesso tradicionais. Estes três fatores de decisão podem ser usados para o desenvolvimento de vários modelos mais detalhados.

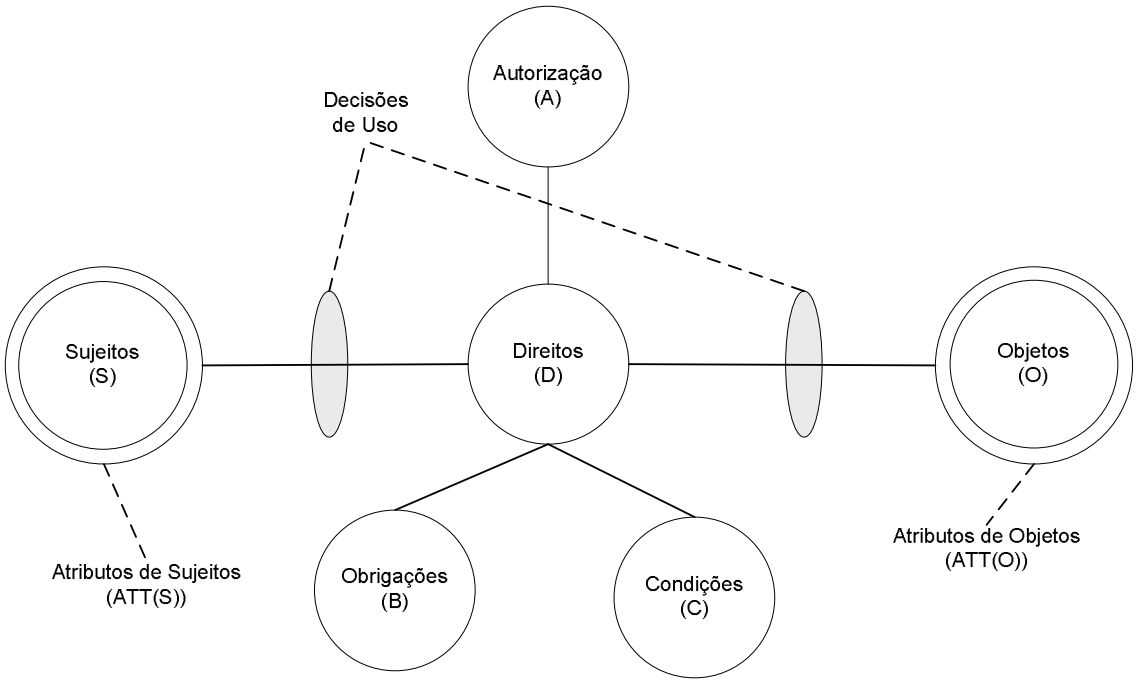


Figura 4.6: Componentes do Modelo $UCON_{ABC}$

Um sujeito é uma entidade associada a atributos e possui ou exerce certos direitos nos objetos. Por exemplo, pode se considerar um ser humano como um sujeito no UCON. Um sujeito é definido e representado por seus atributos. Os atributos do sujeitos são propriedades ou capacidades que podem ser usadas no processo de decisão de uso. Identificação, nomes de grupos, associações e papéis, entre outros, são exemplos de atributos. Os atributos são

divididos em mutáveis, aqueles que podem sofrer modificações causadas pelo acesso, e imutáveis, que só podem ser modificados por ações administrativas.

Objetos são conjuntos de entidades sobre as quais os sujeitos possuem direitos. Os objetos são associados a atributos e a direitos. Assim como nos sujeitos, os atributos dos objetos possuem algumas propriedades que podem ser utilizadas nas decisões de acesso. Exemplos de atributos de objetos são rótulos de segurança, dono do objeto e classes, entre outros. O objeto também pode ter atributos mutáveis.

Direitos são privilégios que um sujeito pode possuir e exercer em um objeto. Os direitos são compostos de conjuntos de funções que habilitam o acesso do sujeito no objeto. O conceito de direito no UCON é essencialmente similar ao mesmo conceito no controle de acesso tradicional. A única diferença é que no UCON os direitos não são vistos isoladamente em uma matriz de controle de acesso independentes da atividade do sujeito.

Autorizações são propriedades funcionais que devem ser avaliadas na decisão de uso, indicando se o sujeito (requisitante) tem permissão para exercer os direitos solicitados no objeto. As autorizações são avaliadas com base nos atributos de sujeito, atributos de objeto, e requisições de direitos. Autorizações podem ser classificadas em pré-autorizações (preA) ou autorizações contínuas (onA). As autorizações do tipo preA são executadas antes da validação da solicitação de um direito enquanto que as autorizações do tipo onA são executadas no exercício do direito.

Obrigações são propriedades funcionais que verificam quais requisitos obrigatórios um sujeito deve executar antes ou durante o uso de um objeto. Assim como as autorizações, as obrigações podem ser divididas em pré-obrigações (preB) e obrigações contínuas (onB).

Condições são fatores de decisão referentes ao ambiente ou orientadas a sistemas. Podem ser usados atributos de sujeitos ou atributos de objeto para selecionar quais exigências de condição devem ser usadas em uma requisição. Alguns exemplos de condição: hora de acesso, local de acesso e estado do sistema.

O modelo UCON pode suportar RBAC. Um papel no modelo RBAC pode ser tratado como um conjunto de usuários e permissões. A permissão é um conjunto de pares objeto-direito. As associações usuário-papel podem ser visualizadas como atributos do sujeito e as associações permissão-papel como atributos de objetos e direitos.

Um exemplo da aplicação do modelo RBAC com o UCON é a realização de uma ação, que necessita do consentimento do paciente, por um médico. A ação só pode ser

realizada pelo médico se o paciente der seu consentimento em um formulário. Esta ação requer um atributo de obrigação adicional que deve ser atendido pelo paciente para que a requisição seja atendida. No exemplo citado a obrigação é do tipo preA, pois deve ser executada antes da liberação do acesso solicitado.

4.3. Outros Modelos de Restrições do RBAC

Vários autores, entre eles Jaeger e Tidswell [JAE00] consideram a limitação de restrições à separação de tarefas no modelo unificado RBAC-NIST razoável e a falta de uma linguagem para especificação como um grande problema do modelo. Para suprir estas duas deficiências surgiram vários trabalhos.

Os trabalhos apresentam basicamente duas propostas: novos tipos de restrições e modelos para descrição formal de restrições. Entre os trabalhos que apresentam novos tipos de restrições os mais relevantes são [KOL03, JAE99, HAN00, TID00a, SAN03, LI03]. As propostas de modelos para descrição formal de restrições mais importantes são [KOL03, CRA03, HAN00, TID00a, TID00b, AHN99, LI03].

Além da restrição de separação de tarefas a proposta de [KOL03] define restrições de pré-requisitos de papéis (*prerequisite roles*), de cardinalidade e um modelo que permite descrição formal de políticas de controle de acesso, análise das políticas e das requisições de controle de acesso. A restrição de pré-requisito de papel, ao contrário da separação de tarefas, define que um usuário pode ser associado a um papel A somente se ele já é membro do papel B ou de um conjunto de papéis. Por exemplo, um usuário só pode ser associado ao papel líder se ele já está associado ao papel especialista ou um usuário só pode sobrescrever um arquivo se o usuário possuir permissão para ler o diretório onde o arquivo se encontra. Cardinalidade refere-se à limitação do número máximo de membros associados a um papel ou ao número máximo de papéis associados a um usuário.

O impacto na complexidade do modelo de restrições do RBAC com a adição de novas características é analisado por [JAE99]. A característica adicionada foi a de papéis sensíveis ao contexto (*context-sensitive roles*). O caso apresentado no artigo é o de uma universidade virtual onde os direitos dos principais dependem do curso no qual eles participam e dos papéis deles nestes cursos. Os instrutores podem criar atividades para os estudantes e materiais de suporte para as atividades. O trabalho mostra que é possível a adição dessa restrição sem ferir os conceitos do RBAC.

Um modelo combinado para especificação e implementação de restrições é proposto por [CRA03]. O modelo permite que, baseado em estruturas de controle de acesso dinâmicas, um monitor de referência baseado em papéis faça validação das restrições durante o acesso.

Por considerar os modelos de especificações de restrições para RBAC incompletos e complicados [HAN00] propôs um *framework* de especificação de restrições bem definidas. O *framework* proposto permite desenvolvedores construírem aplicações com controle de acesso baseado nos papéis dos usuários assegurando que cada papel é configurado com privilégios consistentes, cada usuário é autorizado corretamente aos seus papéis e que cada usuário deve ativar e utilizar seus papéis sem conflitos de interesses. O modelo suporta as restrições de separação de tarefas e cardinalidade.

Uma abordagem matemática integrada para definição de herança e restrições no modelo de controle de acesso dinâmico (*dynamically typed acces control* - DTAC) é apresentada por [TID00a]. A abordagem propõe um modelo de restrições mais simples que o RBAC e com um algoritmo de avaliação mais eficiente que permite especificação de restrições complexas. O artigo demonstra como definir as restrições de separação de tarefas e cardinalidade.

Para expressar política de controle de acesso [TID00b] propõe a utilização de um modelo gráfico no qual os nós representam conjuntos (usuários, objetos, etc.) e as setas representam as relações entre os conjuntos. As restrições são definidas por um conjunto pequeno de operadores incluídos nas associações entre os nós do gráfico. A modelagem de restrições de separação de tarefas é demonstrada na proposta.

Ahn e Sandhu [AHN99] descreveram um *framework* para especificação de políticas de separação de tarefas e conflitos de interesses em sistemas baseados em papéis. Para especificar tais políticas é proposta uma linguagem formal intuitiva na qual funções de sistema e conjuntos são os elementos básicos. A especificação da separação de tarefas é demonstrada utilizando a linguagem.

Um *framework* (RT) que combina as melhores características do RBAC e dos sistemas de gerenciamento de confiança (*trust-management systems*), para a definição de políticas, é apresentado por [LI03]. O RT possui as noções de papéis, usuários, permissões, sessão e ativação seletiva de papéis do RBAC e dos sistemas de gerenciamento de confiança possui os princípios de gerenciamento distribuído de autoridade através do uso de credenciais. O artigo demonstra como definir a política de restrições de separação de tarefas através do *framework*

utilizando papéis múltiplos (*manifold roles*). O conceito de papéis múltiplos é apresentado para substituir a forma tradicional de separação de tarefas. Um papel múltiplo é composto por um conjunto de papéis e só pode ser ativado com o consentimento dos principais associados aos papéis do conjunto.

4.4. Conclusões do Capítulo

Este capítulo apresentou diversos modelos de restrições para o RBAC. O modelo de restrições do modelo Unificado RBAC-NIST aborda apenas restrições de separação de tarefas, o modelo $UCON_{ABC}$ apresenta uma abordagem nova e mais ampla para as restrições e as outras propostas analisadas apresentam outros tipos de restrições como pré-requisito de papéis, cardinalidade e papéis múltiplos.

Para que haja evolução no estudo de restrições do RBAC é necessária a definição de um padrão para especificação de restrições. Cada modelo estudado apresenta sua própria forma de especificação, o que dificulta a proposta de novos trabalhos nesta área.

Dos vários modelos analisados o modelo UCON é o mais flexível para definição de restrições, pois ele permite restrições de autorização, obrigação e condição. Uma limitação do modelo unificado RBAC-NIST é a abordagem somente de restrições de separação de tarefas. O modelo de restrições do RBAC não suporta, por exemplo, a especificação de políticas em ambientes onde é necessária a concordância de um conjunto de principais para a realização de uma tarefa. Tarefas com este tipo de necessidades estão geralmente associadas a atividades não convencionais (críticas), por exemplo, envolvendo emergência médica ou um ambiente de segurança máxima. Uma forma de adequar o RBAC às atividades não convencionais é estendê-lo baseado no modelo $UCON_{ABC}$.

No capítulo seguinte será mostrada a proposta desta dissertação utilizando o conceito de pré-obrigações do Modelo UCONABC no modelo de restrições do RBAC.

Capítulo 5

Proposta e Aspectos de Implementação

O modelo de restrições do RBAC não atende satisfatoriamente as necessidades de um ambiente com tarefas não convencionais (críticas) que não podem ser subdivididas em um conjunto de subtarefas e executadas seqüencialmente. Por exemplo, um general precisa acessar informações confidenciais que dependem de pré-obrigações para serem visualizadas. Estas pré-obrigações podem exigir, por imposição do modelo, consenso de outros generais para serem acessadas.

Está se propondo estender o modelo RBAC com restrições do modelo unificado RBAC-NIST, apresentado na seção anterior, para suportar tarefas não convencionais como a descrita acima. A proposta é permitir a criação de papéis especiais que demandem a autorização de um conjunto pré-definido de papéis para serem ativados. Assim, é possível especificar restrições definindo a necessidade de aprovação de um quorum mínimo de principais para que uma determinada tarefa crítica seja executada por um principal.

Um exemplo de tarefa crítica é a abertura de um cofre de segurança máxima de uma instituição financeira, onde 2 profissionais (um gerentes e um tesoureiro) um de cada lado da porta que dá acesso ao cofre, devem girar simultaneamente as chaves em seu poder para abrir a porta do respectivo cofre.

5.1. Extensão do Modelo RBAC com Restrições

Como comentado anteriormente, o modelo RBAC não prevê obrigações. Para dar suporte a tal situação, está se propondo a adição de um novo tipo de papel ao RBAC (Figura 5.1), o papel especial. Além de estar associado a um conjunto de permissões, o papel especial, indica – através de restrições – quais papéis simples (papéis que um principal pode ativar

sozinho) precisam autorizar sua ativação. A Figura 5.1 mostra o diagrama de classes conceitual do modelo RBAC com a extensão proposta. O diagrama apresentado a seguir tem como base o diagrama apresentado na seção 4.1.3. Algumas classes e relacionamentos foram retirados para facilitar a visualização.

A extensão proposta adiciona duas especializações de classe ao modelo original: Papel Especial e Restrição de Papel Especial. As restrições para ativação de um Papel Especial em uma sessão foram divididas em duas: Restrições onB e restrições preB. As restrições preB são aplicadas quando uma sessão é inicializada com a ativação de um Papel Especial, enquanto as restrições onB são aplicadas quando um Papel Especial precisa ser ativado durante uma sessão. Na ativação de um Papel Especial é instanciada a classe Restrição de Ativação de Papel Especial que indica os papéis que precisam concordar com a ativação do Papel Especial.

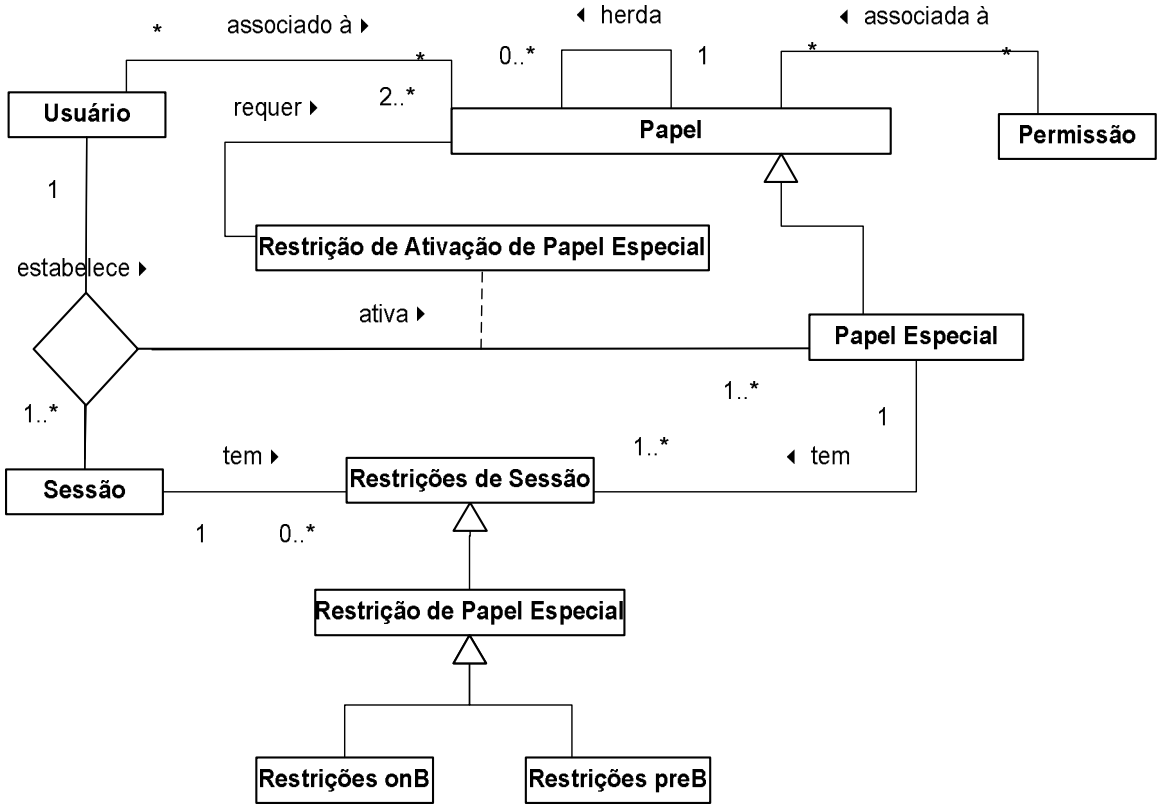


Figura 5.1: Diagrama de classes com a extensão proposta

Se o cofre do exemplo citado anteriormente suportasse abertura eletrônica, através do RBAC seria possível automatizar sua abertura com o uso do papel especial (Figura 5.2). Supondo que um papel especial, “operador da porta do cofre”, tenha permissões para executar a tarefa abrir a porta do cofre, sendo as restrições para sua ativação a autorização dos principais gerente e tesoureiro. Então, se ambos os profissionais Maria e João (Figura 5.2), através de seus papéis autorizarem a ativação do papel do operador, o cofre poderá ser aberto. Porém, nem João nem Maria poderiam fazê-lo sozinhos.

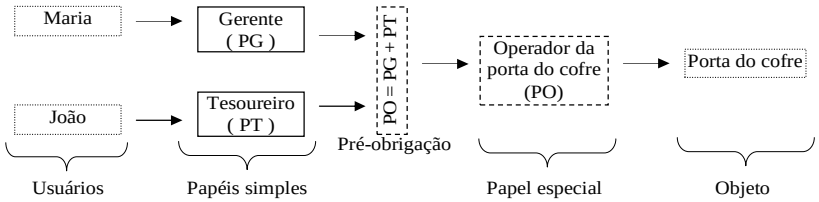


Figura 5.2: Exemplo de ativação de um papel especial

Para dar suporte aos papéis especiais no modelo RBAC está se propondo a adição de um atributo a cada papel do RBAC. Este atributo será denominado nível de acesso (NA) e armazenará um valor numérico diferente para cada tipo de papel no sistema. Foi atribuído um valor numérico ao NA para se poder representar o nível de acesso dos papéis especiais – que é gerado a partir da adição dos NAs dos papéis simples necessários para a ativação.

A adição de NAs especificada nos papéis especiais pode resultar em valores redundantes e gerar falhas de segurança, logo, será utilizado como NA de papéis simples valores do sistema de numeração binário resultantes da expressão 2^n (potência de 2). Papéis simples podem assumir valores como: $1(0001_b)$, $2(0010_b)$, $4(0100_b)$, $8(1000_b)$, etc. Papéis especiais podem assumir NAs com valores que representam a operação OU/*OR* (|) binário dos papéis simples envolvidos, por exemplo: NA=3 (0001_b | 0010_b = 0011_b), NA=6 (0010_b | 0100_b = 0110_b), etc. Assim, os papéis especiais, são facilmente identificados pois, NA=6 por exemplo, só pode ser resultante da ativação do bit de ordem 2 (2^2) e do bit de ordem 1 (2^1). Significando que o papel de NA=6 só pode ser ativado com as autorizações dos papéis simples de NA = 2 e de NA = 4. Observe que a estratégia de numeração de papéis simples, adotada, permite-nos considerar os papéis especiais uma soma de tais papéis, efetivada através da ativação posicional dos bits do respectivo NA, como mostra a Figura 5.3.

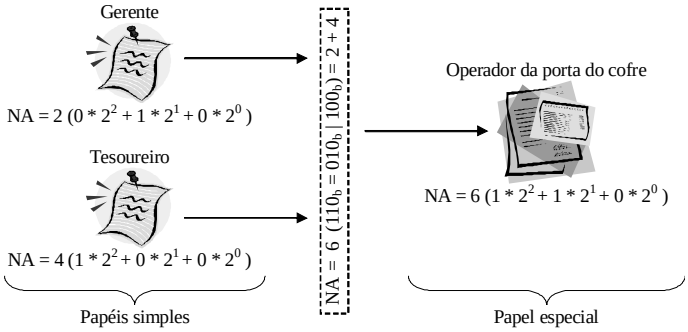


Figura 5.3: Ativação de papel especial

Retomando o exemplo da abertura do cofre citado acima, atribuindo-se ao "operador da porta do cofre" NA = 6 (Figura 5.3), pode-se concluir que este representa um NA composto por outros NAs, como exposto anteriormente. Considerando então, o gerente com NA = 2 e o tesoureiro com NA = 4 e, que o papel "operador da porta do cofre", está associado a ambos, sabe-se que nenhum pode abrir a porta a sós. Porém, por exemplo, se um gerente (NA = 2) iniciar uma sessão e tentar ativar o papel "operador da porta do cofre", poderá fazê-lo se obtiver a autorização de um tesoureiro. De acordo com o comportamento descrito anteriormente para este caso, o controlador RBAC ativará o papel de operador da porta (NA = 6) e a sessão do gerente terá autorização para abrir a porta do cofre (Figura 5.3).

Todas as ações envolvendo a ativação de papéis especiais gerarão registros de auditoria (adicionais), que são acrescidos aos dados comumente encontrados neste tipo de registro, pois, todos os dados dos participantes da sessão também serão registrados. O período de validade de um papel especial ativo numa sessão deve ser definido pelos principais que autorizam a ativação do mesmo.

Na Figura 5.4 é mostrado o diagrama de seqüência especificando o modelo de restrições baseado em pré-obrigações proposto neste trabalho. Neste caso, as pré-obrigações representadas pelos NAs são aplicadas apenas na ativação dos papéis no estabelecimento de sessão; esta limitação é derivada do modelo unificado RBAC-NIST. Porém, as pré-obrigações propostas podem ser aplicadas em qualquer momento da sessão onde a ativação de um papel se fizer necessária.

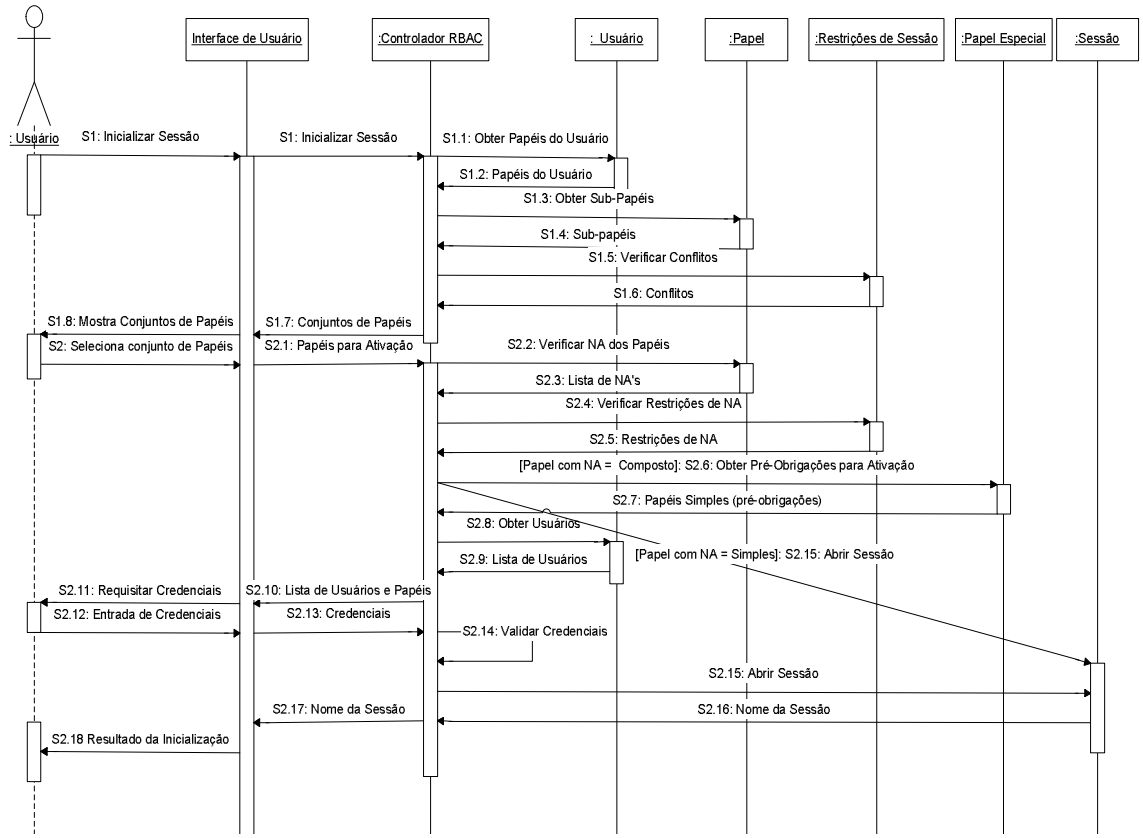


Figura 5.4: Diagrama de seqüência com as extensões propostas

Na dinâmica de estabelecimento da sessão representada na Figura 5.4, o usuário solicita a inicialização da sessão, no evento S1 – através da interface de usuário – que acionará o controlador RBAC. Após recuperar os papéis associados ao usuário, o controlador RBAC mostra ao mesmo o conjunto de papéis disponíveis para a ativação (evento S1.8). No evento S2 são impostas as pré-obrigações propostas neste trabalho e no evento S2.11 a autorização necessária (conjunto de credenciais) para a abertura da sessão é requisitada ao usuário. Quando as credenciais são fornecidas e verificadas com sucesso pelo controlador RBAC, a sessão é aberta no evento S2.15 (Figura 5.4) e o usuário é informado do fato no evento S2.18. Em caso contrário, o usuário é informado da falha na inicialização da sessão através do mesmo evento S2.18.

Todas as extensões propostas neste trabalho estão em conformidade com a especificação vigentes do RBAC.

A seguir será descrito um cenário que ilustra o uso das extensões propostas para suportar as necessidades de um ambiente de emergências médicas. Nesta proposta não é

necessário considerar regras de relaxamento da política de autorização do sistema para atender as demandas do mesmo. Observa-se que a aplicação pode ser qualquer, o ambiente de emergências médicas apenas pareceu mais intuitivo para ilustrar a aplicação das extensões propostas.

5.2. Estudo de Caso

5.2.1. Cenário

Quando um paciente chega inconsciente ao setor de emergência de um hospital, o profissional que for atendê-lo deverá ter acesso a informações privadas do mesmo. Segundo a legislação e a ética médica, os dados do paciente são confidenciais. Porém, o controle de acesso não pode prejudicar o atendimento do paciente, negando acesso legítimo às informações e serviços requisitados por pessoal médico.

Considere que o usuário E (Figura 5.5), um paciente, tenha sofrido algum tipo de trauma e esteja desacordo no setor de emergência de um hospital, utilizando um sistema com as extensões propostas neste trabalho. O médico plantonista (B), NA=4, para ter acesso ao prontuário do paciente com um nível de permissões equivalentes ao “médico do paciente” (NA=6) precisaria, por exemplo, de uma credencial de um membro da família (NA=2). Neste caso, observa-se que o NA=6 (papel especial) e o NA=32 (médico do paciente), tem nível de acesso diferente mas direitos equivalentes. Esta estratégia foi adotada para permitir uma maior grau de independência na gestão das políticas de autorização e mais clareza para a auditoria. Evidentemente, outras combinações de NAs, concedendo diferentes níveis de acesso ao prontuário poderiam ser geradas.

No exemplo foi considerada também a hipótese do médico necessitar acesso a informações confidenciais – as quais só o paciente teria acesso; no prontuário do paciente poderia constatar que o mesmo possui uma determinada doença. Se esta informação chegasse ao conhecimento do público poderia causar dano à imagem do paciente. Neste caso, para obter permissões equivalentes ao paciente seria necessário, além de um familiar, a presença de um representante legal do mesmo, um advogado por exemplo. Assim, o advogado participando da autorização, permitiria que o papel de NA=14 (papel especial) fosse ativado.

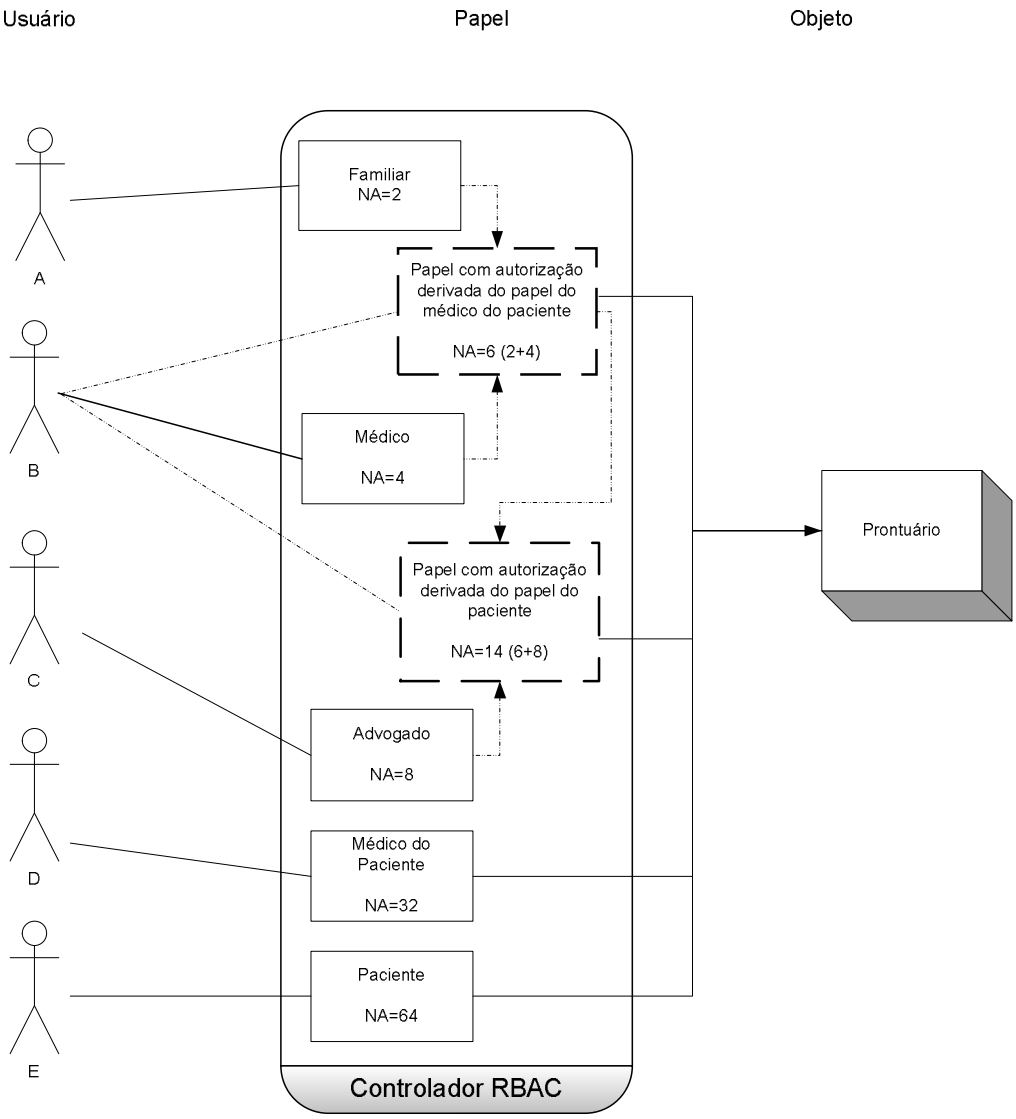


Figura 5.5: Ativação de papel especial em um hospital

É possível perceber pelo exemplo que podem ser atribuídos graus de acesso diferentes a um objeto, associando lhes diferentes papéis especiais. Neste caso, cada papel especial podendo demandar um número diferente de principais e permitindo diferentes privilégios de acesso. É evidente, neste caso, que a participação dos principais na autorização representa apenas um endosso para a ativação do papel especial.

Em nenhum dos casos, onde usuários adicionais são exigidos (no exemplo, familiar e advogado) é necessária a presença física do mesmo no hospital – pode-se utilizar qualquer técnica conhecida de credenciamento de terceiros para o exercício dos respectivos papéis.

5.2.2. Trabalhos relacionados

Talvez, o principal obstáculo para o uso de Prontuário Eletrônico do Paciente (PEP) sejam as questões relacionadas à segurança e confidencialidade dos dados do paciente. Vários autores têm discutido e apresentado várias propostas envolvendo o assunto.

Nos casos de emergências médicas, basicamente, há duas tendências com relação ao controle de acesso. Alguns autores baseiam suas propostas em fatores circunstanciais e outros no relaxamento do controle de acesso como alternativas para contorna as exigências específicas do ambiente.

Em [MOT01, BEZ98] é proposto o tratamento dos casos de emergência baseado em fatores circunstanciais, tais como a localização do usuário, relacionamento usuário/paciente e data/hora de acesso para limitar o uso da situação de emergência médica a casos pré-definidos.

Os trabalhos de [MOR97, LON00, CHA01] apontam para o relaxamento do controle de acesso e para o registro da ocorrência em uma área onde os registros não podem ser apagados. Ou seja, o mecanismo de controle de acesso deve permitir acesso quando for sinalizada uma situação de emergência e registrado o fato. A abordagem proposta por [LON00] sugere o envio de uma notificação eletrônica para todas as partes interessadas no momento que a situação de emergência é definida. Em [CHA01] a situação é tratada por um mecanismo mais complexo, definido como forma forte de autenticação baseada em dois fatores (*stronger form of authentication like two-factor authentication*). Esse mecanismo atua na requisição de uma sessão de emergência verificando se o papel que o usuário desempenha está mapeado como um papel que pode ativar tal tipo de sessão. Se existir o mapeamento, o acesso é liberado e a operação é registrada em uma área segura diferente da área do sistema.

A principal desvantagem destas abordagens é que a definição da situação como uma emergência médica, cabe a um usuário (o operador do sistema, por exemplo). Logo, se cria uma brecha de segurança no sistema porque um usuário pode entrar no sistema como se a situação fosse de emergência e obter dados privilegiados sem que na verdade a situação caracterize uma emergência médica. Todos os mecanismos comentados acima – que registram de alguma forma o fato – pretendem limitar o uso desta situação especial para coibir sua utilização indevida. Porém, na verdade tais registros tornam-se tão corriqueiros para os envolvidos na situação que ao longo do tempo perdem importância e podem passar desapercebidos ou notados muito tempo depois que foi feito seu mau uso.

A partir da abordagem proposta neste trabalho situações como as descritas acima não deverão ocorrer porque o acesso só é permitido com autorização de um grupo de principais. Logo, cabe aos interessados (paciente, familiares, representantes legais, etc.) a decisão de autorizar, ou não, o acesso as informações privilegiadas do prontuário. Ao contrário dos casos anteriores – onde muitas vezes o paciente ou as pessoas de seu relacionamento podem nem ter ciência que as informações de seu prontuário vazaram – em nossa abordagem o dono do prontuário, o paciente, é que decide quem está autorizado a permitir o acesso ao seu prontuário em sua incapacidade de fazê-lo.

5.3. Aspectos de implementação

A proposta foi efetivada com base na implementação RBAC/Web [FER99] do NIST. Esta implementação é uma aplicação intranet, onde o RBAC é utilizado como esquema de autorização para controlar o acesso às páginas de um servidor http. Na implementação RBAC/Web, usuários correspondem a logins no servidor. Transações HTTP que podem ser executadas por usuários (através dos seus papéis) nas páginas html representam as permissões RBAC. O controle de acesso do RBAC/Web foi desenvolvido numa API (implementada em C e Perl) e os códigos são de domínio público.

O protótipo simula o ambiente descrito no cenário (seção 5.2), onde o prontuário representa uma página html com as informações do paciente. A efetivação do protótipo foi alcançada fazendo-se adaptações nas classes da implementação RBAC/Web, como mostram as Tabelas 5.1 e 5.2. Na Tabela 5.1, são descritos brevemente os métodos administrativos e, na Tabela 5.2, os métodos de suporte ao sistema; houve casos em que novos métodos precisaram ser implementados.

Foi incluído na implementação RBAC/Web o método privado writeAccessLevel na classe Role. Esse método é executado no momento da criação do papel pelo método addRole e grava o papel e seu nível de acesso no arquivo NA_role para arquivar a relação entre papéis a seus níveis de acesso. Na exclusão de um papel (DeleteRole) seu nível de acesso é marcado como inativo ao invés de ser apagado; o NA é preservado pelo método rmAccessLevel para fins de auditoria futuros e nunca é reutilizado no sistema.

Tabela 5.1: Funções administrativas afetadas pela implementação da proposta

Métodos	Descrição
AddRole (Alterado)	Incluir novos papéis
DeleteRole (Alterado)	Excluir papéis
writeAccessLevel (Incluído)	Gravar o papel e seu nível de acesso
rmAccessLevel (Incluído)	Excluir um nível de acesso

Tabela 5.2: Funções de suporte afetadas pela implementação da proposta

Método	Descrição
CreateSession (Alterado)	Criar uma sessão de usuário com um conjunto padrão de papéis ativos
checkAccessLevel (Incluído)	Verificar o nível de acesso
writeAccessLog (Incluído)	Gravar log de ativação de papel especial
create_login_special_choices	Criar menu com papéis especiais que o usuário pode ativar
create_login_choices	Criar menu com conjunto de papéis simples que o usuário pode ativar
split_special_roles	Criar conjunto de papéis e usuários que precisam concordar com a ativação de um papel especial
checkSpecialRole	Validar a credencial dos usuários que concordam com a ativação de um papel especial

Para o estabelecimento de uma sessão a página de login utiliza o método CreateSession da classe Session que invoca os métodos privados: create_login_choices (cria o conjunto de papéis que o usuário pode ativar – que não possuem separação dinâmica de tarefas) e writeARS (grava os papéis selecionados pelo usuário no arquivo nome_usuario.active_roles). O método create_login_choices foi alterado para não mostrar os papéis especiais. O novo método create_login_special_choices, que mostra os papéis especiais que o usuário pode ativar, é invocado logo após o método create_login_choices. Na Figura 5.6 pode-se observar o menu para ativação de papéis. A opção *Select a Role Set for Session* permite ao usuário selecionar um conjunto de papéis normais que deseja ativar enquanto que a opção *Select a Special Role for Session* permite o usuário escolher um papel especial. Após a escolha do papel a ser ativado o método writeARS seria chamado, mas foi preciso adicionar um método para verificar se o papel que está sendo ativado é especial ou não. Antes da execução do método writeARS é invocado o método checkAccessLevel. Este método foi incluído para verificar se o papel que o usuário está tentando ativar é especial.

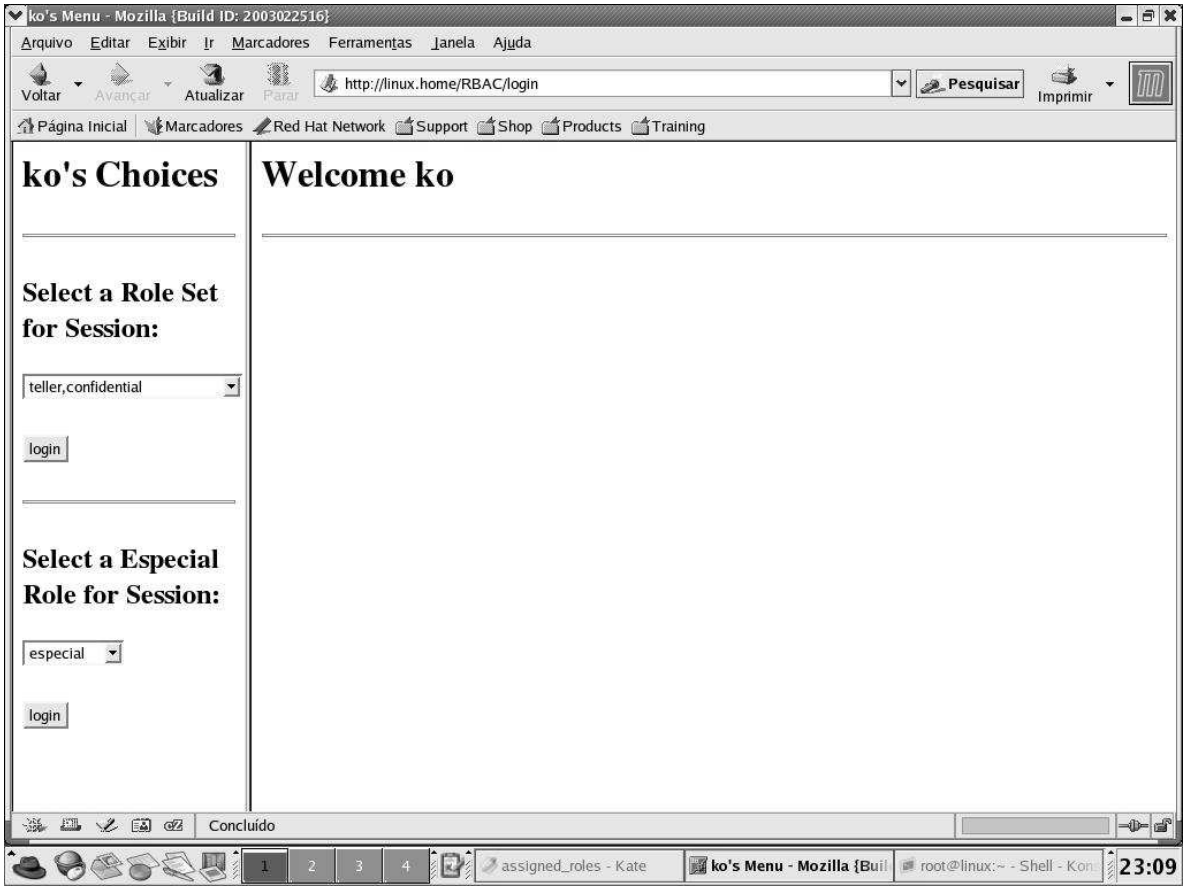


Figura 5.6: Tela de escolha de para ativação

Quando a ativação for de um papel especial o método split_special_roles é invocado para mostrar o conjunto de papéis e usuários que precisam concordar com a ativação do papel especial (Figura 5.7). O método checkSpecialRole verifica a existência das credenciais dos principais requeridos para a ativação; as credenciais são buscadas em arquivos no caso do protótipo. Após a validação das credenciais o método writeARS é executado. Quando um papel especial é ativado o método writeAccessLog grava informações adicionais ao log, como data/hora da ativação, papel especial ativado e principais/papéis que autorizaram a ativação.

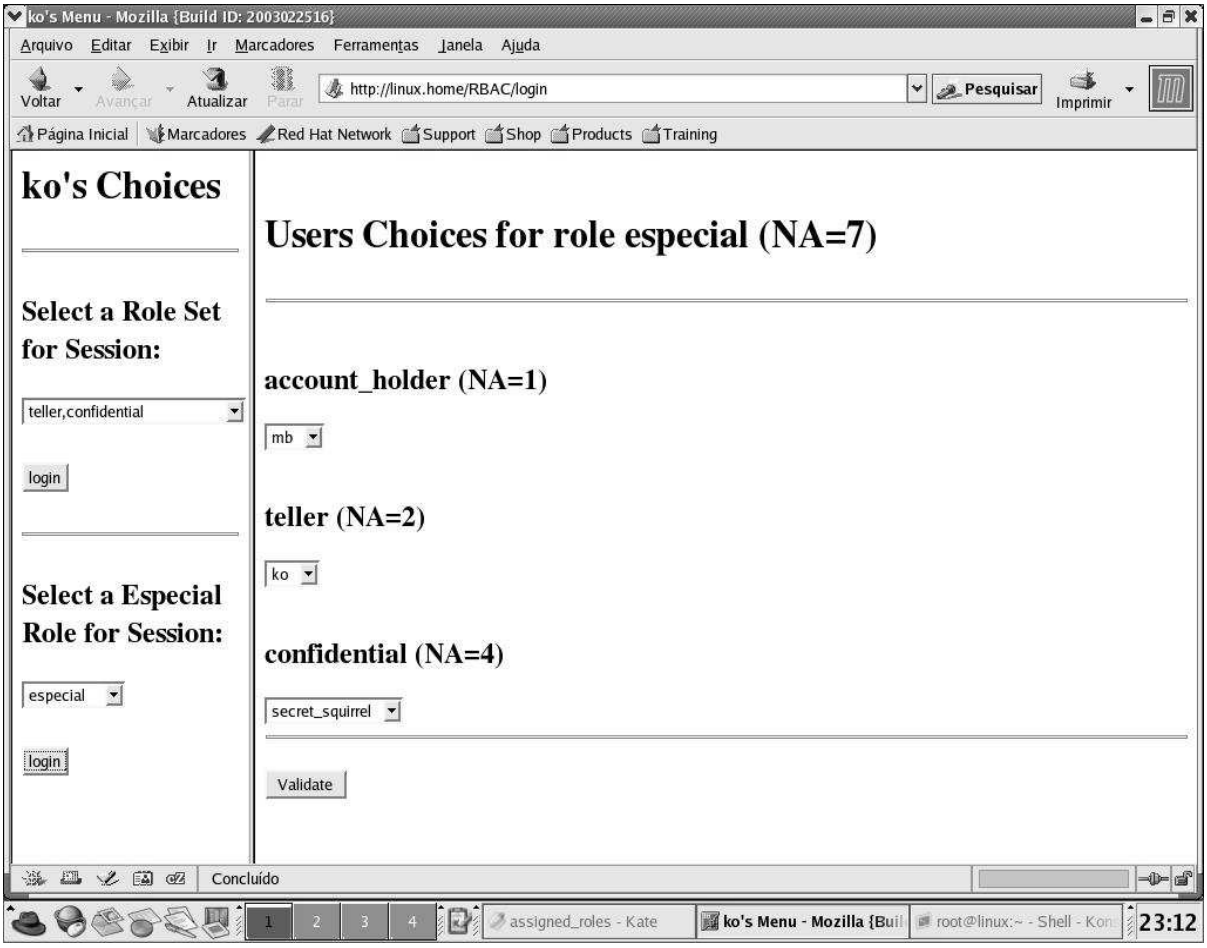


Figura 5.7: Tela de usuários para ativação de um papel especial

5.4. Testes de Performance do Protótipo

Foram aplicados testes de desempenho nos métodos que montam o menu com as opções de papéis que o usuário pode ativar (create_login_choices e create_login_special_choices) e no método que mostra o conjunto de papéis e usuários que precisam concordar com a ativação de um papel especial (split_special_roles).

Para testar a performance do menu com as opções de papéis que o usuário pode ativar a metodologia usada foi aumentar o número de papéis associados ao usuário e medir o tempo de execução dos métodos create_login_choices e create_login_special_choices. Na tabela 5.3 pode-se visualizar o número de papéis simples e especiais associados a um usuário e o tempo médio para montar o menu.

Tabela 5.3: Tempo médio para criar menu de login

Número de papéis simples	Número de papéis especiais	Tempo
5	0	1 s
10	0	1 s
15	0	2 s
20	0	14 s
25	0	Mais de 5 min
5	5	1 s
10	5	1 s
15	5	2 s
20	5	14 s
25	5	Mais de 5 min

O tempo para montagem do menu começa aumentar consideravelmente com 20 papéis simples associados ao usuário e a adição de papéis especiais não causa impacto no tempo. O aumento no tempo é causado pelo método que verifica a separação de tarefas e monta o conjunto de papéis simples que o usuário pode ativar.

Nos testes de performance do método split_special_roles, que mostra o conjunto de papéis e usuários que precisam concordar com a ativação de um papel especial, a metodologia utilizada foi aumentar o número de papéis simples necessários para concordar com a ativação de um papel especial e aumentar o número de usuários associados aos papéis simples. O maior tempo de execução do método split_special_roles foi de 2s quando o papel especial a ser ativado precisava da concordância de 20 papéis simples. Cada papel simples tinha 40 usuários associados.

5.5. Conclusões do Capítulo

A extensões ao modelo de restrição do RBAC propostas permitem o suporte a situações críticas respeitando a política de autorização do sistema.

A aplicação RBAC/Web permitiu implementar um protótipo para validar a proposta. O protótipo implementado mostrou-se efetivo na realização do controle de acesso através da ativação de papéis especiais. As credencias dos principais (papéis simples), foram implementadas em arquivos para a efetivação dos testes, mas poderiam ser implementadas através de certificados digitais ou mesmo do esquema usuário/senha. As alterações efetuadas na aplicação causaram pouco impacto na performance do sistema.

Um aspecto importante não abordado no trabalho foi a implementação de métodos para revisão de nível de acesso. Pelo menos duas funções de revisão são importantes. Uma função de revisão de nível de acesso “papel especial”–“papel simples” onde a partir de um papel especial seja possível relacionar todos os NAs de papéis simples e respectivos usuários ligados ao mesmo. E uma função de revisão “papel simples”–“papel especial” que permita à partir de papéis simples obter todos os NAs de papéis especiais a que o referido papel simples está associado.

Capítulo 6

Conclusão

Este trabalho apresentou uma proposta de extensão do modelo RBAC com as restrições do modelo ABC. Apesar do RBAC ter agregado as melhores características dos modelos clássicos (discricionário e obrigatório) no sentido de ser flexível como o primeiro e centralizado como o segundo, o modelo de restrições do RBAC é baseado no princípio da separação de tarefas suportado pelo princípio do mínimo privilégio. Assim, a definição de políticas de autorização é fortemente influenciada pelo modelo de restrições. Outros tipos de restrições foram considerados e discutidos pelos estudiosos da área, mas não foram adotadas no modelo de referência.

A divisão de uma tarefa em subtarefas, executadas em seqüência, garante a conformidade com o modelo de restrição do RBAC, mas restringe a especificação de políticas. Neste modelo não é possível a especificação de políticas em ambientes onde é necessária a concordância de um conjunto de principais para a realização de uma tarefa. Tarefas com este tipo de necessidade estão geralmente associadas a atividades não convencionais (críticas), por exemplo, envolvendo emergência médica ou um ambiente de segurança máxima.

Nos trabalhos relacionados, considerando situações de emergência médica, tal situação é considerada uma exceção. Então, as regras da política de segurança são relaxadas em prol da vida do paciente e o médico plantonista tem acesso a todos os dados do paciente [MOT01, BEZ98, MOR97, LON00, CHA01]. Esta abordagem não parece muito coerente porque permite de certa forma a criação de uma brecha no sistema de segurança, que pode ser mal utilizada.

Este trabalho apresentou uma proposta de extensão ao modelo de restrição do RBAC para suportar situações críticas como as descritas acima, respeitando as regras da política de autorização do sistema. A extensão proposta permite a criação de papéis (denominados papéis especiais) que só podem ser ativados com a autorização de um conjunto pré-definido de papéis simples. As atividades críticas são associadas aos papéis especiais. Ou seja, para que um principal realize uma atividade crítica, o mesmo deve ativar um papel especial que, por sua vez, necessita da aprovação de um quorum mínimo de principais para ser ativado. Os principais que participam da aprovação agem no sentido de aprovar a ativação do papel especial e não com o objetivo de exercer os direitos associados aos seus papéis.

Existem diversas possibilidades de continuidade deste trabalho. Duas possibilidades são mais evidentes. A primeira é o desenvolvimento de funções para revisão de nível de acesso “papel especial”–“papel simples” e “papel simples”–“papel especial”. E a segunda propor a utilização de uma forma padronizada para especificação de políticas de controle de acesso.

As funções de revisão de nível de acesso podem ser implementadas como métodos administrativos. A função de revisão de nível de acesso “papel especial”–“papel simples”, a partir de um papel especial deverá relacionar todos os NAs de papéis simples e respectivos usuários ligados ao mesmo. A partir de papéis simples, as funções de revisão “papel simples”–“papel especial” deverá fornecer todos os NAs de papéis especiais a que o referido papel simples está associado.

Uma alternativa interessante a ser analisada para especificação de políticas é o XACML, que se encontra em desenvolvimento. Existe um esboço em estágio avançado para se especificar políticas RBAC no modelo XACML.

Referências Bibliográficas

[AHN99] AHN, G.-J.; SANDHU, R. *The RSL99 Language for Role-Based Separation of Duty Constraints*. In Proceedings of 4th ACM Workshop on Role-Based Access Control, Fairfax, VA, Oct. 1999, p. 43-54.

[AMO94] AMOROSO, E. G. *Fundamentals of Computer Security Technology*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1994, ISBN: 0-13-108929-3.

[BAR97] BARKLEY, J.; FERRAIOLO, D. *Specifying and Managing Role-Based Access Control within a Corporate Intranet*. RBAC 97 Fairfax, Va, USA, 1997, p. 77.

[BEL73] BELL, D. E.; LA PADULA, L. J. *Secure Computer Systems : Mathematical Foundations*. MTR-2547 Vol. I, The MITRE Corporation, Bedford, Massachusetts, Mar. 1973.

[BEZ98] BEZNOSOV, K. *Requirements for Access Control: US Health-care Domain*. Proc. of the 3rd ACM Workshop on Role-based Access, Fairfax, VA, USA, 1998.

[BIB77] BIBA, K. J. *Integrity Considerations for Secure Computer Systems*. Technical Report ESD-TR-76-372, USAF Electronic Systems Division, Hanscom Air Force Base, Bedford, Massachusetts, Apr. 1977.

[BRE89] BREWER, D.; NASH, M. *The Chinese wall security policy*. In Proceedings of the Symposium on Security and Privacy, IEEE Press, Los Alamitos, Calif., 1989, p. 215–228.

[CHA98] CHANDRAMOULI, R.; SANDHU, R. *Role-Based Access Control Features in Comercial Database Management Systems*. In Proceedings of 21st NIST-NCSC National Information Systems Security Conference, Arlington, VA, Oct. 1998, p. 503-511.

[CHA01] CHANDRAMOULI, R. *A Framework for Multiple Authorization Types in a Healthcare Application System*. In: Proceedings. of the 17th ACSAC, IEEE, 2001.

[CLA87] CLARK, D.; WILSON, D. *A comparasion of commercial and military computer security policies*. Proceedings of the IEEE Computer Society Simposium of Research in Security and Privacy, Los Alamitos, Calif., 1987, p. 184-194.

[CRA03] CRAMPTON, J. *Specifying and Enforcing Constraints in Role-Based Access Control*. In Proceedings of 8th ACM Symposium on Access Control Models and Technologies (SACMAT03), Como, Italy, Jun. 2003, p. 43-50.

[DEN76] DENNING, D. *A Lattice Model of Secure Information Flow*. Communication of ACM, Vol. 19, Nº. 5, May 1976.

[DEN82] DENNING, D. *Criptography and data security.* Addison-Wesley, 1982.

[FER92] FERRAIOLO D., KUHN, R. *Role-Based Access Control*. Proc. Of the 15th NIST - NCSC National Computer Security Conference, Baltimore, Oct. 1992.

[FER92b] FERRAIOLO, D.; GILBERT, D.; LYNCH, N. *Assessing Federal and Commercial Information Security Needs.* NISTIR 4976, National Institute of Standards and Technology, Gaithersburg, MD, Nov. 1992.

[FER95] FERRAIOLO, D.; CUGINI, J.; KUHN, R. *Role Based Access Control: Features and Motivations*. In proceedings of 11th Anual Computer Security Applications Conference, New Orleans, LA, Dec. 1995, p. 241-248.

[FER99] FERRAIOLO, D.F.; BARKLEY, J.F.; KUHN R. *A Role Based Access Control Model and Reference Implementation within a Corporate Intranet*. Proc.of. National Institute of Standars and Technology, Vol. 2, nº. 1, February 1999, p. 34-64.

[FER01] FERRAIOLO, D. F.; SANDHU, R.; GAVRILA, S.; KUHN, D. R.; CHANDRAMOULI, R. *A Proposed Standard for Role Based Access Control*. ACM Transactions on Information and System Security, New York, Volume. 4, No. 3, ago. 2001, p. 224-274.

[FRI97] FRIBERG, C.; HELD, A. *Support for Discretionary Role Based Access Control in ACL-oriented Operating Systems*. RBAC 97 Fairfax, Va, USA, 1997, p. 83.

[GAV98] GAVRILA, S.I.; BARKLEY, J.F. *Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management*. Proc. of 3rd ACM Workshop on Role-Based Acess, Fairfax, VA, 1998, p. 81-90.

[GLE99] GLENN, F. *RBAC in UNIX Administration*, Proceedings of 4th ACM Workshop on Role-Based Access Control, Fairfax, VA, Oct. 1999, p. 95-101.

[GOG82] GOGUEN, J.A;. MESAJUER, J. *Security Policies And Security Models*. Proceedings of IEEE symposium on Reseach in Security and Privacy, 1982.

[GUI95] GUIRI, L. *A new model for role-based access control*. In proceedings of the 11th Annual Computer Security Application Conference, New Orleans, LA, Dec. 11-15 1995, p. 249-255.

[HAN00] HAN, Y.; CHUN-GEN X.; GONG-XUAN, Z.; FENG-YU, L. *Constraint specification for object model of access control based on role*. ACM SIGSOFT Software Engineering Notes, Volume 25 , No 2, Mar. 2000, p. 60-63.

[JAE99] JAEGER, T. *On the Increasing Importance of Constraints*. In Proceedings of 4th ACM Workshop on Role-Based Access Control, Fairfax, VA, Oct. 1999, p 33-42.

[JAE00] JAEGER, T. *Rebuttal to the NIST RBAC Model Proposal*. In proceedings of the 5th ACM Workshop on Role-Based Access Control. Berlim, Germany, 2000.

[JAN98] JANSEN, W.A. *A Revised Model For Role-Based Access Control*. NIST IR 6192, Techical Report, NIST, Jul. 9, 1998.

[KOL03] KOLACZEK, G. *Specification and Verification of Constraints in Role Based Access Control for Enterprise Security System*. In proceedings of 12th IEEE International Workshops on Enabling Technologies (WETICE 2003), Infrastructure for Collaborative Enterprises, Linz, Austria, Jun 9-11, 2003.

[KUH97] KUHN D.R. *Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems*.,Proc. of 2nd ACM Workshop on Role-Based Access Control, 1997, Virginia, USA, p. 23-30.

[LAM71] LAMPSON, B.W. *Protection*. Proc. 5th Princeton Conf. on Information Sciences and Systems, Princeton, p. 437, 1971.

[LAN83] LANDWEHR, C. E. *Best available technologies for computer security*. IEEE Computer Vol. 16, No. 7, Jul. 1983, p. 86-100.

[LAN84] LANDWEHR, C.E.; HEITMEYER, C.L.; MCLEAN, J. *A security Model for Military Message Systems*. ACM Transactions on Computer Systems, Vol. 2, No 3, Aug. 1984, p. 198-222.

[LI03] LI, N.; MITCHELL, J. C. *RT: A Role-based Trust-management Framework*. In Proceedings of The Third DARPA Information Survivability Conference and Exposition (DISCEX III), Washington, D.C., Apr. 2003. IEEE Computer Society Press, Los Alamitos, California, p. 201-212.

[LON00] LONGSTAFF, J.J; LOCKYER, M.A.; CAPPER, G.; THICK, M.G. *A model of accountability, confidentiality and override for healthcare and other applications.* Proceedings of 5th ACM Workshop on Role- Based Acess Control, Berlin, Germany, 2000.

[MAC97] MACKENZIE, D.E.; POTTINGER, G. *Mathematics, Technology, and Trust: Formal Verification,* Computer Security, and the U.S. Military, IEEE Annals of the History of Computing, Vol. 19, No 3, 1997.

[MCL90] MCLEAN, J. *The Specification and Modeling of Computer Security*. IEEE Computer, Vol. 23, No 1.

[MOR97] MORGER,O., NITSCHE, U.; TEUFEL, S. *Security Concerns for Mobile Information Systems in HealthCare*. In: Proceedings of the 8th International Conference and Workshop on Database and Expert Systems Applications DEXA'97, IEEE Society, Los Alamitos, 1997.

[MOT01] MOTTA, G.; FURUIE, S. *Autorização e controle de acesso para o prontuário eletrônico do paciente em ambientes abertos e distribuídos: uma proposta de modelo e arquitetura*. In: WSeg'2001 - Workshop em Segurança de Sistemas Computacionais, Florianópolis, SC, Feb. 2001.

[NYA99] NYACHAMA, M.; OSBORN, S. *The Role Graph Model and Conflict of Interest*. ACM Transactions on Information and System Security, Vol 2, no. 1, Feb. 1999. p. 3-33.

[OBE01] OBELHEIRO, R.R.; WESTPHALL, C.M; FRAGA, J.S. *Controle de Acesso Baseado em Papéis para o Modelo CORBA de Segurança*. In: Anais do 19o. Simpósio Brasileiro de Redes de Computadores (SBRC'2001), Florianópolis, SC, May 2001, p. 869-884.

[ORA92] ORACLE CORPORATION. *Oracle 7 Server SQL Language Reference Manual*. Dec. 1992.

[OSB00] OSBORN, S.; SANDHU, R.; MUNAWER, Q. *Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies*. ACM Transactions on Information and System Security, Vol. 3, no. 2, 2000.

[PAR02] PARK, J.; SANDHU, R. *Towards Usage Control Models: Beyond Traditional Access Control*. In Proceedings of 7th ACM Symposium on Access Control Models and Technologies, Naval Postgraduate School, Monterey, California, USA. ACM, Jun. 2002, p. 57-64

[PAR04] PARK, J.; SANDHU, R. *The $UCON_{ABC}$ Usage Control Model*. ACM Transactions on Information and System Security, Vol. 7, No. 1, Feb. 2004, p. 128–174.

[SAN00] SANDHU, R.; FERRAIOLO, D.; KUHN, D.R. *The NIST Model for Role-Based Access Control: Towards A Unified Standard*. Proc. of 5th ACM Workshop on Role-Based Acess Control, Berlin, Germany, 2000.

[SAN03] SANDHU, R.; PARK, J. *Usage Control: A vision for Next Generation Access Control*, In The Second International Workshop Mathematical Methods, Models and Architectures for Computer Networks Security (MMM-ACNS), St. Petersburg, Russia, Sep. 2003.

[SAN96] SANDHU, R.; COYNE, E.J.; FEINSTEIN, H.L.; YOUMAN, C.E. *Role-Based Access Control Models*. IEEE Computer, Vol. 29, Nº. 2, Feb. 1996, p. 38-47.

[SAN98a] SANDHU R. *Role Activations Hierarchies*. In Proceedings of 3rd ACM Workshop on Role Based Acess Control. Fairfax, VA, Oct. 1998.

[SAN98b] SANDHU, R. *Role-Based Access Control*. In Advances in Computers , Volume 46. Academic Press, 1998.

[SAN99] SANDHU, R.; BHAMIDIPATI, V.; MUNAWER, Q. *The ARBAC97 Model for Role-Based Administration of Roles*. ACM Transactions on Information and System Security, Vol. 2, nº 1, Feb. 1999.

[TID00a] TIDSWELL, J.; JAEGER, T. *Integrated Constraints and Inheritance in DTAC*. In proceedings of the 5th ACM Workshop on Role-Based Access Control, Berlim, Germany, 2000

[TID00b] TIDSWELL, J.; JAEGER, T. *An Access Control Model for Simplifying Constraint Expression.* In proceedings of the 7th ACM conference on Computer and communications security, Athens, Greece, 2000, p.154-163.

[WES00] WESTPHALL, C.M. *Um esquema de autorização para a segurança em sistemas distribuídos de larga escala*. Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina, Tese de doutorado, Florianópolis – Santa Catarina, 2000.