Capítulo

4

A Nova Geração de Modelos de Controle de Acesso em Sistemas Computacionais

Luiz Otávio Botelho Lento[1], Joni da Silva Fraga[2] e Lau Cheuk Lung[3]

1 Depto. Automação de Sistemas, Universidade Federal de Santa Catarina,– otavio@das.ufsc.br
2 Depto. Automação de Sistemas, Universidade Federal de Santa Catarina - fraga@das.ufsc.br
3 PPGIA, Pontifícia Universidade Católica do Paraná - lau@ppgia.pucpr.br

Abstract

The access control is a security service and is used to manage the access to the resources (ex data, process and devices) from a computer system, making the actions or operations of a valid user may execute limited. In the last few years, the access control has been improving itself, your representation and functions. This short course has the objective to present a few models of access control, such as DAC, MAC, DRM, RBAC and UCON, approaching the meaning characteristics and proprieties, and also a brief comparison between then.

Resumo

O controle de acesso é um serviço de segurança, e tem como função gerenciar o acesso aos recursos (dados, processos, dispositivos, etc.) de um sistema computacional, limitando as ações ou operações que um usuário válido possa executar. Nos últimos anos, o controle de acesso vem evoluindo quanto a sua representação e funcionalidades. Este mini-curso visa apresentar alguns modelos de controle de acesso, como DAC, MAC, DRM, RBAC e UCON, abordando as suas principais características e propriedades, como também um breve comparativo entre eles.

4.1. Introdução

O controle de acesso que teve no modelo Matriz de Acesso a sua principal expressão formal quando introduzido em 1971 por Butler W. Lampson [Lampson 1971], tem sido mantido como definitivo na descrição de políticas simples. Grande parte das políticas discricionárias são descritas na simplicidade deste modelo formando os controles conhecidos na literatura como controles discricionários (DAC – *Discretionary Access Controls*). Alguns autores identificam ainda estes controles como baseados em identidades [Karp, A. H. (2006)] (IBAC – *Identity-Based Access Control*) devido ao fato que os controles que implementam as políticas, dependem fortemente da autenticação dos sujeitos que solicitam os acessos controlados.

Os mesmos modelos que descrevem políticas obrigatórias (Modelos MAC – *Mandatory Access Control*) incorporam a simplicidade do Matriz de Acesso nas verificações em um mesmo nível de segurança. Modelos como o Bell-LaPadula [Bell e LaPadula 1976], que trata da confidencialidade, e o Biba, que se fundamenta na verificação da integridade dos acessos, são exemplos deste grupo de modelos. Estes modelos são também identificados como baseados em regras (*Rule-Based Access Model*) porque envolvem a concretização de políticas globais em um sistema ou organização. As regras de acesso definem a perda do caráter essencialmente discricionário dos criadores dos objetos acessados, e possibilitam a verificação de propriedades mais globais, que devem valer no sistema como um todo.

Nos últimos anos, o controle de acesso vem experimentando uma evolução acentuada com novas formas de representação e também pela necessidade de adequação a novas aplicações e tecnologias. Um exemplo, destes novos modelos é o RBAC (*Role-Based Access Control* - controle de acesso baseado em papel) [Sandhu 1997] que simplifica a gerência de direitos por não mais concentrar sob identidades os mesmos, mas sim em papéis ou funções. O controle de acesso motivado pelos direitos de propriedade e o controle de uso também provocaram o aparecimento de modelos como o DRM (*Digital Right Management*) [Ku e Chi 2004]. Recentemente, Sandhu e Park propuseram o UCON (*Usage Control Model*) [Sandhu. e Park 2004] como expressão máxima de modelos e políticas que pode englobar todos os modelos citados, tradicionais ou não. Este modelo geral é uma ferramenta poderosa tanto para a formalização como para a implementação de todos os tipos de controles em um sistema.

Este mini-curso tem como objetivo central apresentar os modelos citados em suas principais características e propriedades e verificar suas utilidades na formalização e implementação dos diferentes controles de acesso identificados na literatura. Este documento está dividido em uma parte que trata com modelos mais convencionais, onde são apresentados modelos como DAC, MAC, DRM e RBAC. Uma boa parte deste texto explora ainda o modelo de controle de acesso UCON. Uma outra parte que trata alguns modelos de controle de acesso desenvolvido por pesquisadores. Por fim, o texto termina com uma análise comparativa entre estes modelos descritos onde procuramos evidenciar a importância de cada modelo na expressão dos controles de sistemas e aplicações.

4.2. Aspectos Básicos sobre Segurança em Sistemas Computacionais

Na seqüência foram colocadas algumas definições envolvendo Segurança, Política e Modelos de Segurança que usamos no texto. Não são objetos do mini-curso políticas e controles externos que visam à proteção dos equipamentos e sistemas computacionais. No final desta seção caracterizamos ainda controle de acesso.

4.2.1. Conceitos

Segurança

A segurança em sistemas computacionais não é formada exclusivamente por meios que visam proteger informações ou recursos computacionais, mas é, antes de tudo, uma disciplina que através de seus conceitos, metodologias e técnicas, tenta manter propriedades de um sistema, evitando ações danosas no mesmo. Na literatura, existem várias definições para Segurança (*security*) e, em quase todas, a mesma é caracterizada como a qualidade de serviço que visa manter no sistema um conjunto de propriedades [Landwehr 2001, Denning 1982]:

- A **Confidencialidade** garante a revelação da informação só a sujeitos autorizados. [1]
- A **Integridade** assegura a não modificação indevida – seja acidental ou intencionalmente – das informações e recursos no sistema.
- A **Disponibilidade** garante que as informações e recursos num sistema computacional estarão desimpedidos e prontos para serem usados quando requisitados por sujeitos autorizados.

Alguns autores ainda juntam às citadas, as propriedades de Autenticidade e de Não Repúdio [Landwehr 2001]. A legitimidade de informações e de principais é explicitada pela propriedade de autenticidade. O não repúdio garante, em protocolos e transações, as proteções contra comportamentos omissos ou maliciosos onde participantes neguem ações realizadas.

As Violações de segurança em sistemas computacionais correspondem a burlar de alguma forma a segurança de um sistema computacional de modo a não se verificarem uma ou mais propriedades de segurança. A Tabela 4.1 ilustra os tipos de violação em contraposição às propriedades de segurança não verificadas.

Tipo de Violação		Propriedade de Segurança Violada
1	Revelação Não Autorizada	Confidencialidade
2	Modificação Não Autorizada	Integridade
3	Negação de Serviço	Disponibilidade

Tabela 4.1. Tabela de relação de referências.

As violações de segurança são decorrências de **vulnerabilidades** (*vulnerability*), **ameaças** (*threat*) e **ataques** (*attack*) em sistemas computacionais. Entende-se por vulnerabilidades, as fraquezas ou imperfeições em procedimentos, serviços ou sistemas,

[1] Entende-se por sujeito uma entidade ativa como um humano, sistema ou máquina.

oriundas de falhas de concepção, implementação ou de configuração dos mesmos. Uma ameaça é a caracterização de um possível conjunto de ações que explore as vulnerabilidades e o conhecimento sobre um sistema que possa por em risco as propriedades de segurança. Uma ameaça, quando concretizada na execução de suas ações, é identificada como um ataque à segurança do sistema.

Políticas de Segurança

O termo *política de segurança* pode ter significados diferentes dependendo do nível em que se aplica. Em ambientes computacionais, política de segurança é entendida normalmente como um conjunto de regras que especificam como um sistema provê os seus serviços, mantendo as propriedades de confidencialidade, integridade e de disponibilidade [Lendweir2001]. Os sistemas computacionais fazem então uso de regras através de *controles*, estabelecendo os limites de operação dos usuários no sistema e protegendo seus dados e recursos da ação de intrusos[2]. Uma política sempre se aplica a um sistema específico, e não a uma classe geral de sistemas.

As políticas de segurança são classificadas em duas categorias: as discricionárias e obrigatórias. Nas discricionárias os acessos a cada recurso ou informação são manipulados livremente pelo proprietário ou responsável pelo mesmo, segundo a sua vontade (à sua discrição). Já nas obrigatórias (não discricionárias) 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 [Mackenzie 1997]. Neste texto, serão ainda tratadas as políticas discricionárias e não discricionárias.

Modelos de Segurança

Os modelos de segurança correspondem a descrições formais do comportamento de um sistema atuando segundo regras de uma política de segurança. Estes modelos são representados na forma de um conjunto de entidades e relacionamentos [Goguen 1982]. A definição de políticas de segurança é normalmente orientada por modelos de segurança, que fornecem na representação abstrata o funcionamento seguro do uso no sistema alvo de um conjunto de regras de segurança.

Os modelos se apresentam, na literatura, divididos em três tipos básicos [Sandhu, e Samarati 1996]:

- Controles baseados em identidade ou discricionários (*Discretionary Access Control*: DAC): por expressarem as políticas discricionárias, baseiam-se na idéia de que o proprietário do recurso deve determinar quem tem acesso ao mesmo.

[2] As regras definidas pelas políticas de segurança determinam as entidades autorizadas e responsáveis pelas ações executadas sobre informações mantidas no sistema, normalmente identificadas como **principal** (sujeitos autorizados). O nível de aplicação desta política pode caracterizar um principal como um usuário, um processo ou ainda uma máquina em uma rede de computadores. A entidade (usuário, processo ou máquina) que ganha acesso a recursos de um sistema computacional violando a política de segurança é normalmente denominada de **intruso**.

- Controles baseados em regras gerais ou obrigatórios (*Mandatory Access Control*: MAC): baseiam-se em uma administração centralizada de segurança, na qual são ditadas regras incontornáveis de acesso à informação. 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 (vide seção 4.3.2).
- Controles baseados em papéis (*role*). (*Role-Based Access Control*—RBAC): requer que permissões de acesso sejam atribuídas a papéis e não a usuários, como no DAC; os usuários obtêm estes direitos através de papéis alocados a si.

Mecanismos de Segurança

Os mecanismos de segurança são responsáveis pela concretização das políticas de segurança nos sistemas computacionais. Estas políticas, cujos comportamentos são expressos através de modelos de segurança, são implantadas por mecanismos que exercem os controles necessários para manter as propriedades de segurança. Os controles executados internamente em sistemas computacionais que gerenciam os acessos a recursos são identificados como **Controles de Acesso**. Controles usados na proteção das informações que são disponíveis através de dispositivos de entrada e saída (memórias secundárias, suportes de comunicação, etc.), envolvem o que é normalmente identificado como **Controles Criptográficos**. Outros controles ainda podem ser identificados em sistemas computacionais. Estes controles são chamados de Serviços de Autenticação, importantes na identificação de principais (sujeitos autorizados), e Controles de Inferência, que normalmente envolvem as semânticas das aplicações. O estudo aqui apresentado está centrado em Controle de Acesso.

4.2.2. Controle de Acesso

O controle de acesso limita as ações ou operações que um sujeito de um sistema computacional pode executar, restringindo o que ele pode fazer diretamente, como também os programas que podem ser executados em seu nome. A Figura 4.1 apresenta o esquema básico do controle de acesso exercido através de mecanismos em um sistema computacional.

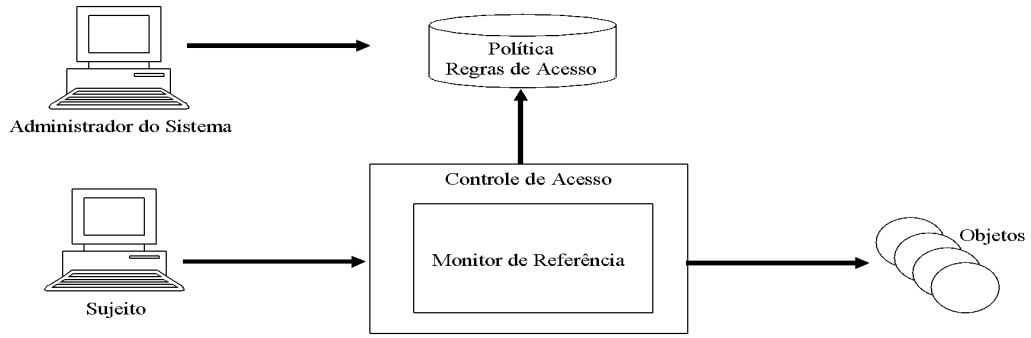

Figura 4.1. Controle de acesso

Na efetivação do controle de acesso são usados mecanismos que tomam o nome de **Monitor de Referências** [Anderson 1972], e que atuam em vários níveis de um sistema. As referências a segmentos de memória são validadas nas camadas inferiores do sistema, através do *hardware*. O sistema operacional, por sua vez, através do seu serviço de arquivos valida os acessos a arquivos no sistema. O monitor de referência é o mediador de toda a tentativa de acesso de sujeitos aos objetos do sistema, consultando as regras da política para verificar se as solicitações de acesso são permitidas. As regras são mantidas pelo administrador de segurança (ou de sistema), tendo como base uma política de segurança.

O monitor de referência como responsável na intermediação de todas as requisições de acesso a objetos de um sistema resultou na definição de **núcleo de segurança** [Landwehr 1983]. Este núcleo que envolve um conjunto de mecanismos de hardware e software, permitindo a concretização da noção de um monitor de referências, deve possuir algumas propriedades: ser inviolável, incontornável (sempre ativado nas requisições de acesso) e pequeno o suficiente para permitir a verificação de sua correção. A idéia de núcleo de segurança deu origem mais tarde as ***TCBs***[3] (*Trusted Computing Base*) introduzidas pelos critérios de avaliação do DoD [DOD85] (também conhecidos como *Orange Book*).

Os aspectos envolvendo a implementação destes monitores de referência (ou na sua versão de núcleo de segurança) não são muito evidentes. Por exemplo, na definição das permissões de acesso, as abordagens existentes tentam atingir soluções comuns que atendam a seu grupo de principais e recursos controlados. O grau de refinamento e a simplicidade de gerenciamento são metas a serem também consideradas. Todavia, a abordagem usada, quando da definição das permissões, certamente não conseguirá levar em consideração todos os aspectos existentes, e prestigiará sempre um determinado grupo dos recursos controlados.

Os sistemas de controle de acesso podem ser diferenciados via suas políticas e seus mecanismos de acesso. As políticas de acesso são direcionamentos de alto nível, baseadas nas necessidades dos proprietários dos recursos ou ainda das organizações. A partir destas definições de alto nível devem ser geradas permissões que determinam como os acessos serão controlados em todos os níveis do sistema.

4.3. Modelos de Segurança

A pesquisa na área de modelos de segurança computacional começou no início da década de 70. Ao longo desses anos, inúmeros modelos foram propostos, com as mais variadas premissas e os mais diversos objetivos. Esta seção apresenta detalhes de modelos de segurança considerados clássicos: Matriz de Acesso, Bell-LaPadula, Biba e RBAC. Na seqüência destes, são apresentadas as novas proposições de modelos presentes na literatura.

[3] Uma *Trusted Computing Base* também deve ter as mesmas propriedades de um núcleo, consistindo na concentração da totalidade dos mecanismos de segurança de sistema computacional (incluindo hardware, software). A combinação destes mecanismos é responsável em cumprir a política de segurança.

4.3.1. Matriz de Acesso Segurança

O modelo de segurança utilizado em boa parte dos sistemas atuais é o chamado controle de acesso discricionário (DAC), que delega aos usuários a tarefa de proteger seus recursos no sistema. Neste modelo, cada usuário é quem determina quais os direitos de acesso que outros usuários ou aplicações do sistema possuem sobre as informações que são de sua responsabilidade. Modelos discricionários são considerados inadequados para diversas aplicações, uma vez que são relativamente fracos e demasiadamente flexíveis: basta um equívoco por parte de um usuário inocente (ou um ato deliberado de um usuário malicioso) e informações importantes podem ser indevidamente reveladas, alteradas ou destruídas.

O modelo de **Matriz de Acesso** [Sandhu e Samarati 1994] é o modelo conceitual subjacente ao controle de acesso discricionário. Neste modelo, o estado de segurança do sistema é representado pela tripla (S, O, A), onde: S é o conjunto de sujeitos s_i que podem exercer privilégios; O é um conjunto de objetos o_j nos quais os privilégios ou direitos podem ser exercidos; e A é a matriz de acesso onde linhas correspondem aos sujeitos em S e as colunas aos objetos em O. Uma célula A_{ij} da matriz representa os direitos de acesso do sujeito s_i sobre o objeto o_j. É importante ressaltar que sujeitos podem ser também objetos. Por exemplo, um dos acessos representados na matriz pode ser o envio de um sinal a um processo; neste caso, os sujeitos correspondentes a processos deveriam ser incluídos também nas colunas da matriz.

A Figura 4.2 apresenta um exemplo da matriz de acesso com três objetos (três arquivos) e três sujeitos (usuários Waldir, Nanda e Luiz). Os direitos sobre arquivos são os usuais: dono (D), leitura (L), escrita (W) e execução (E). A entrada *A[Waldir, Arquivo1]* representa os privilégios D, L, W e E de Waldir sobre o Arquivo 1.

	Arquivo 1	**Arquivo 2**	**Arquivo 3**	**Arquivo 4**
Waldir	D W, L, E		D W, L, E	L
Nanda	E	D W, L	E	
Luiz	E	L		D W, L

Figura 4.2. Matriz de Acesso

No controle de acesso discricionário, a concessão e a revogação dos direitos de acesso a um objeto são feitas pelo usuário que é dono desse objeto (à sua *discrição*). Isso fornece ao usuário uma grande flexibilidade na proteção de seus objetos, o que é uma vantagem deste modelo de controle de acesso. Entretanto, o controle de acesso discricionário não permite controlar a disseminação da informação. No exemplo da figura 4.3, Nanda permite que Luiz leia o seu arquivo 2, mas proíbe que estas informações sejam lidas por Valdir. Entretanto, não há nada no modelo que possa impedir que Luiz aja maliciosamente, copiando as informações do arquivo 2 para 0 arquivo 4, o que possibilitaria que Valdir lesse tais informações mesmo contra a vontade da usuária Nanda.

A matriz de acesso proposta por Lampson [Lampson 1971] é um modelo relativamente informal. O modelo de matriz de acesso é definido em termos de estados de segurança do sistema. A matriz corresponde ao estado atual de segurança do sistema. As mudanças de estado de segurança do sistema são realizadas usando **regras de transição do modelo** [Landwehr 1981]. Estas correspondem a permissões para mudar o objeto "matriz" e são tipicamente **retirar sujeito/objeto**, **criar sujeito/objeto**, **transferir direitos** e **suprimir direitos.** A aplicação destas regras determina situações de acessos especiais:

> Um sujeito pode acessar um objeto porque possui o direito no estado de segurança atual do sistema.
>
> Ou, um sujeito pode acessar um objeto porque pode obter o direito necessário através de mudanças de estado da matriz.

Um estado não autorizado ou estado de fuga é aquele onde um direito pode ser obtido por um sujeito não autorizado. Portanto um sujeito não autorizado a aceder um objeto pelo seu proprietário pode por mudanças de estado da matriz vir a obter o direito de acesso necessário ao objeto. Este aspecto de possíveis evoluções em tempo de execução da matriz e, por conseqüência, do estado de segurança do sistema, determinaram o aparecimento de várias extensões formais e gráficas do modelo matriz de acesso para verificar este comportamento dinâmico do modelo. Exemplos destas extensões são os modelos formais *HRU* [Harrison 1976] e *Take-Grant* [Snyder 1981] que fundamentados no modelo de matriz de acesso, estudam a disseminação de direitos devido a evolução dinâmica do estado de segurança de um sistema.

A implementação da Matriz de Acesso na forma original em grandes sistemas torna-se inviável devido ao seu tamanho, e a grande quantidade de células em branco. Existem duas abordagens tradicionais para a implementação do modelo Matrizes de acesso: as listas de controle de acesso (*ACLs*) e listas de competências (*lists of capabilities*).

Listas de controle de acesso (*Access Control Lists*: *ACLs*)

Esta é, talvez, a abordagem mais popular de implementação da matriz de acesso. Cada objeto é associados uma ACL que indica os sujeitos no sistema com acessos autorizados ao objeto considerado. Uma ACL corresponde no armazenamento da matriz por colunas. A Figura 4.3 apresenta um exemplo de ACL.

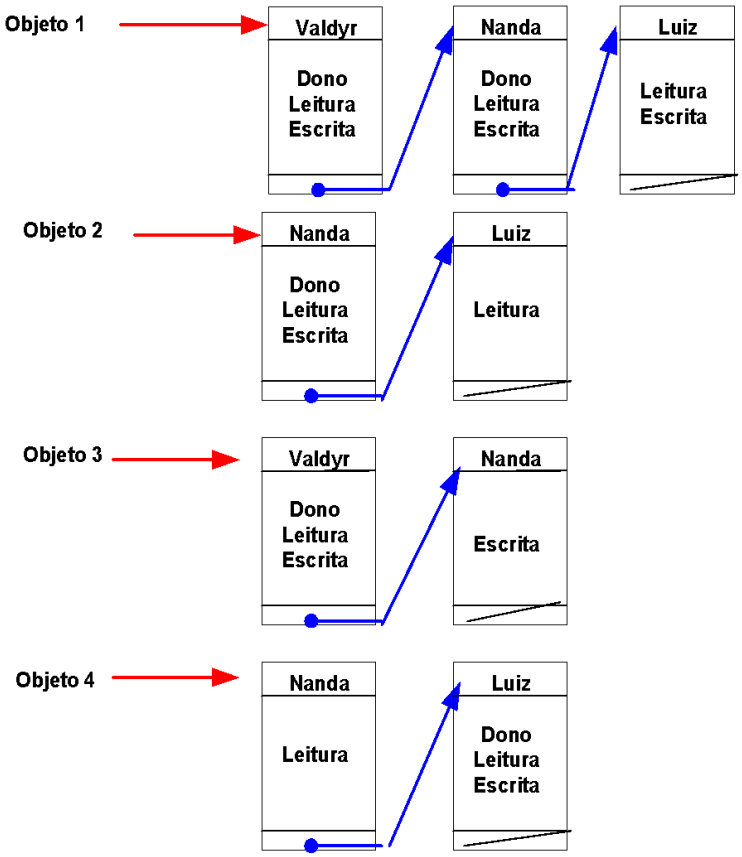

Figura 4.3. ACL

A *ACL* facilita determinar quais os modos de acesso que os sujeitos estão autorizados em um objeto, provendo uma forma fácil de rever ou revogar os modos de acessos aos objetos. Porém, é difícil determinar todos os acessos que um sujeito possui, porque seria necessário verificar a ACL de cada objeto. As listas de acesso são mecanismos normalmente usados em níveis altos de um sistema (em nível de usuário).

Listas de Competências (*Capabilities*)

Nesta abordagem, a cada sujeito está associado uma **lista de competências** (*capability list*) que indica, para cada objeto no sistema, quais as permissões de acessos o sujeito possui (armazenamento da matriz de acesso por linhas). Listas de competências permitem fácil verificação e revogação dos acessos autorizados para um determinado sujeito. As vantagens e desvantagens de ACLs e *capabilities* são, como as próprias estratégias, ortogonais entre si.

Uma *capability* corresponde a um identificador protegido (imutável) que identifica o objeto e especifica os direitos de acesso a serem atribuídos ao sujeito possuidor da mesma. Duas propriedades são fundamentais no mecanismo de *capability*:

O *capability* pode ser passada de um sujeito a outro; e

> Nenhum sujeito possuidor de uma *capability* (identificador) pode alterá-la ou construir novas sem uma negociação prévia com TCB (Trusted Computing Base) do sistema.

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 *capabilities* (ou privilégios) e apresente as mesmas quando necessário para obter os acessos desejados; os servidores precisam apenas verificar a validade da *capability* apresentada para liberar o acesso desejado [Sandhu 1994].

A Figura 4.4 apresenta um exemplo de uma lista de ca*pabilities*.

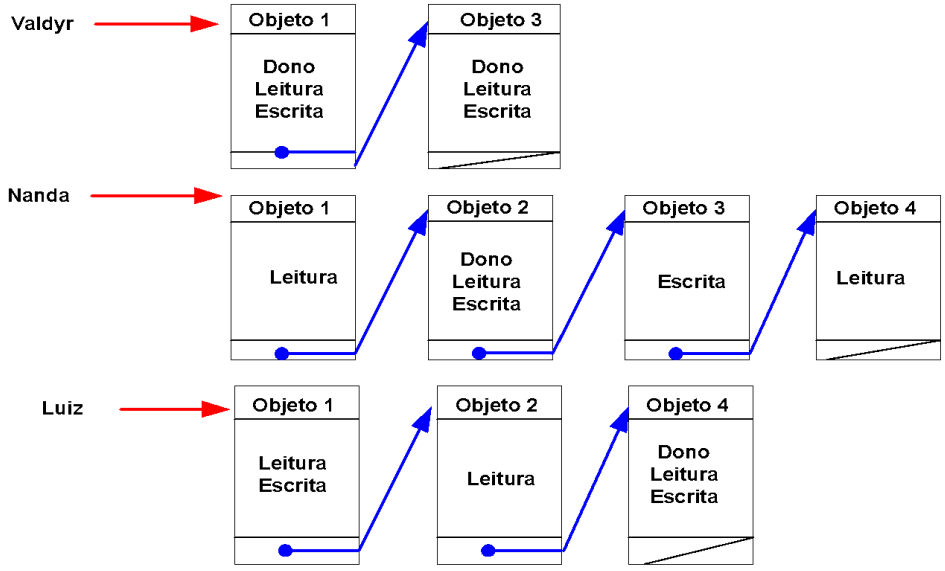

Figura 4.4. Lista de Competências

4.3.2. Modelos Não Discricionários (*Mandatory Access Control* - MAC)

As pesquisas que levaram aos modelos MAC na década de 70, foram financiadas pelo Departamento de Defesa (DoD) dos Estados Unidos. Desta forma, estes modelos iniciais foram baseados em práticas de segurança utilizadas em áreas ligadas à segurança nacional. Em que pese esta origem, os modelos e seus conceitos subjacentes são perfeitamente aplicáveis a ambientes não-militares.

Os controles definidos pelos modelos MAC seguem políticas que valem para todo o sistema e, portanto, que definem regras e estruturas aplicáveis no âmbito de todo o sistema. Estas políticas, normalmente, especificando envolvem algum tipo de classificação multinível (*multilevel policy*) de informação. Os acessos dos sujeitos aos objetos são submetidos a um tipo de controle baseado na classificação usada.

Um dos conceitos importantes nestas classificações usadas em MACs é o de **níveis de sensibilidade**[4]. Como há custos associados à proteção da informação e nem todas as informações são igualmente importantes (ou **sensíveis**), definem-se diferentes níveis de sensibilidade, ordenados segundo uma hierarquia. Os níveis mais usuais são, em ordem crescente de "sensibilidade": NÃO-CLASSIFICADO, CONFIDENCIAL, SECRETO e ULTRA-SECRETO. De maneira similar, uma universidade poderia adotar os níveis ALUNO, FUNCIONÁRIO e PROFESSOR — os níveis devem refletir a necessidade de proteção da informação (dificilmente um ambiente acadêmico classificaria as informações da mesma maneira que um ambiente militar).

Entretanto, a simples associação de níveis de sensibilidade à informação não atende a um princípio clássico de segurança conhecido como *need-to-know*. Este princípio diz que o controle da disseminação da informação está diretamente ligado à quantidade de pessoas que têm acesso a essa informação; desta forma, quanto menos pessoas conhecerem um segredo, mais fácil será garantir que o segredo não será revelado. Para que isso seja viabilizado, são definidas **categorias** (*category*), ou compartimentos de segurança, que correspondem a diferentes projetos ou setores confinando suas informações. Assim, por exemplo, professores do Departamento de Física provavelmente não devem ter acesso a informações classificadas com o nível PROFESSOR pertencentes ao Departamento de Geografia. Os indivíduos podem ter acesso a diferentes categorias na medida em que as suas incumbências demandem este acesso.

Para transpor estes conceitos ao contexto computacional, são definidos **rótulos de segurança** (*security labels*) que agregam os níveis de sensitividade e categorias. Os rótulos de segurança associados as informações de um sistema correspondem ao produto vetorial entre o conjunto de níveis de sensitividade e o conjunto representado por *Category* que definem todos os compartimentos destas informações (*Security label* = *Sensitivity level* x *Category*).

Em sistemas que fazem uso deste mecanismo, todas as entidades recebem um rótulo de segurança; o rótulo de um objeto define a sua **classificação**, e o rótulo de segurança de um sujeito é chamado de **autorização** ou **habilitação** (*clearance*) do sujeito.

Uma das formas de viabilizar a implementação de políticas é construir **reticulados** (*lattice*) com **rótulos de segurança** (*security labels*). Estes reticulados são construídos a partir de uma relação de ordem parcial sobre o conjunto R de rótulos do sistema. Esta relação que permite comparações entre os *security labels* de sujeitos e objetos nestes modelos de políticas multi-nível, é conhecida como relação de dominância e é definida como:

> **Relação de Dominância**: um *Security Level* de um objeto o_1 domina outro de um objeto o_2, se as seguintes condições são verificadas:
>
> $$Sensitivity\_Level(o_1) \geq Sensitivity\_Level(o_2) \wedge Category(o_1) \supseteq Category(o_2)$$

É importante notar que, em conjuntos ordenados parcialmente, existem elementos que são ditos não-comparáveis, e que também, é verificado sempre possui um ínfimo (limite

[4] A tradução mais apropriada para *Sensitivity level* é "nível de sensitividade". Porém acreditamos que "nível de sensibilidade" descreve melhor em Português o aspecto semântico por traz destas classificações e o manteremos neste texto.

inferior) e um supremo (limite superior) segundo a relação de dominância nos reticulados de rótulos de segurança [Landwehr 1981]

Entre os modelos MACs, que fazem uso destas classificações que fazem uso de rótulos de segurança está o modelo Bell-La Padula (BLP) de David Bell e Leonard LaPadula [Bell e LaPadula. 1976] que é descrito a seguir.

O Modelo Bell-LaPadula

Dois cientistas da MITRE Corporation, David Bell e Leonard LaPadula, desenvolveram um modelo baseado nos procedimentos usuais de manipulação de informação em áreas ligadas à segurança nacional americana. Este modelo ficou conhecido como **modelo Bell-LaPadula**, ou modelo BLP [Bell 1976]. Existem diversas outras descrições do modelo Bell-LaPadula disponíveis na literatura, como [Amoroso 1994, Landwehr 1981, Sandhu 1993], algumas delas apresentando pequenas variações em relação ao modelo original. O modelo Bell-LaPadula trata exclusivamente com a confidencialidade das informações no sistema.

Na apresentação original do modelo [Bell 1976], um sistema é descrito através de uma máquina de estados finitos. As transições de estados no sistema obedecem a determinadas regras. Bell e LaPadula demonstram indutivamente que a segurança do sistema é mantida se ele parte de um estado seguro e as únicas transições de estado permitidas são as que conduzem o modelo a um outro estado seguro.

No modelo um 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**, que representa a classificação mais alta dentre as informações já consultadas pelo sujeito no sistema até um determinado instante, sendo, portanto, uma classificação flutuante (dinâmica). A habilitação de um sujeito deve sempre dominar o seu rótulo corrente de segurança.

A **propriedade de segurança simples**, também conhecida como propriedade-ss ou regra *no read up* (NRU)[5], diz que um sujeito só pode observar informações para as quais esteja habilitado; em outras palavras, a **leitura** de um sujeito s_i sobre um objeto o_j é autorizada se, e somente se, *rótulo*(s_i) deve dominar *rótulo*(o_j). Por exemplo, uma informação classificada como SECRETO só pode ser lida por sujeitos com habilitação SECRETO ou ULTRA-SECRETO.

A propriedade-ss não é suficiente para garantir a segurança desejada do sistema: ela não evita que um sujeito malicioso coloque informações privilegiadas em um recipiente com classificação inferior à das informações, o que constitui claramente um fluxo não-autorizado de informação. Assim, torna-se necessário adicionar outra propriedade a ser satisfeita pelo sistema.

[5] *No read up* vem do fato de um sujeito não poder ler objetos localizados acima dele no reticulado de rótulos de segurança.

A **propriedade-*** (propriedade estrela), também chamada de regra *no write down*[6] (NWD), é satisfeita se, quando um sujeito tem simultaneamente um acesso de leitura sobre um objeto o_1 e um acesso de escrita sobre um objeto o_2. Sendo assim, o *rótulo*(o_1) deve dominar o *rótulo*(o_2), isto é, o acesso do sujeito s_i sobre um objeto o_j é autorizado se:

rótulo(o_j) domina o *rótulo-corrente*(s_i) quando o acesso for de escrita;

rótulo(o_j) é dominado pelo *rótulo-corrente*(s_i) quando o acesso for de leitura.

Por exemplo, se um sujeito está lendo um objeto SECRETO, ele só pode alterar um objeto SECRETO ou ULTRA-SECRETO simultaneamente.

Existem duas observações importantes a se fazer respeito da propriedade-*.

1. Ela não se aplica a sujeitos de confiança - um sujeito de confiança é aquele em quem se confia a não transferir informação de modo a quebrar a segurança, mesmo que esta transferência seja possível;
2. Vale a pena lembrar que a propriedade-ss e a propriedade-* devem ser ambas satisfeitas; nenhuma delas garante, por si só, a segurança desejada.

Dinâmica do Modelo Bell-LaPadula

O rótulo corrente de segurança de um sujeito é conceituado como uma classificação flutuante, e define a propriedade-* em termos do rótulo corrente de segurança de um sujeito, sem explicitar como este rótulo efetivamente flutua dentro do sistema. Esta flutuação está ligada ao comportamento dinâmico do modelo BLP, isto é, o rótulo corrente de segurança de um sujeito evolui durante a evolução do próprio sistema.[7]

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 do sistema devem observar a propriedade-ss e a propriedade-*.

Bell e LaPadula [Bell 1976] fornecem um conjunto de regras para a operação de um sistema seguro.[8] Uma destas regras dita que o rótulo corrente de segurança de um sujeito só é modificado mediante uma requisição explícita deste sujeito; isto significa que o rótulo corrente de segurança não flutua de maneira automática no sistema, e, também, que esta flutuação ocorre por iniciativa do próprio sujeito. A regra especifica

[6] Assim chamada porque impede que um sujeito escreva em objetos localizados abaixo dele no reticulado de rótulos de segurança.

[7] O **princípio da tranqüilidade** estabelece que nenhuma operação pode alterar a classificação de objetos ativos no sistema [Landwehr 1981]. Entretanto, implementações baseadas no modelo BLP tipicamente lançam mão de sujeitos de confiança para a reclassificação de objetos.

[8] Evidentemente, a noção de sistema seguro, no contexto do modelo BLP, corresponde a um sistema a salvo de ameaças de revelação não-autorizada.

também que a alteração do rótulo corrente de segurança só é autorizada se ela não violar a propriedade-*.

Por exemplo, seja a seguinte situação: um sujeito s_i, com rótulo corrente NÃO CLASSIFICADO e habilitação (estática) SECRETO, deseja ler um objeto o_1, que é CONFIDENCIAL. A propriedade-ss permite que s_i leia o_1, pois *rótulo*(s_i) domina *rótulo*(o_1). Entretanto, essa operação não satisfaz a propriedade-*, pois *rótulo*(o_1) domina *rótulo-corrente*(s_i). Logo, s_i precisa solicitar a atualização de seu rótulo corrente de segurança para (pelo menos) CONFIDENCIAL. Entretanto, se s_i, ao solicitar a atualização de seu rótulo corrente para CONFIDENCIAL possuir um acesso de escrita para o_2, onde o_2 é igualmente NÃO-CLASSIFICADO, ele deve ter esta solicitação negada pelo sistema, uma vez que a sua aceitação violaria a propriedade-*.

A regra que governa a atualização do rótulo corrente de segurança não impõe qualquer restrição além da satisfação da propriedade-* e da condição de que o rótulo corrente seja dominado pela habilitação do sujeito.

Limitações do Modelo Bell-LaPadula

A adoção do modelo *BLP* pode acarretar problemas se o sistema tiver que lidar também com ameaças de integridade. Quando um sujeito escreve em um objeto com uma classificação superior à sua habilitação (o que satisfaz a propriedade-*), ele não pode observar os efeitos desta operação de escrita (o que violaria a propriedade-ss); por esse motivo, tal operação é chamada de **escrita cega** [Amoroso 1994, Sandhu 1993].

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 objetos com níveis mais altos de segurança podem ser proibidas; um sujeito só poderia escrever em um objeto que tivesse o mesmo nível de segurança. Entretanto, tal modificação restringe, de certa forma, o modelo BLP e muda o 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 da propriedade-* revisada é bastante comum em implementações de sistemas computacionais que seguem o modelo BLP.

O modelo Bell-LaPadula inclui a noção de **sujeitos de confiança** (*trusted subjects*) [Bell 76, Landwehr 1981]. Um sujeito de confiança é aquele em quem se confia a não quebrar a segurança mesmo que alguns dos seus acessos atuais violem a propriedade-*. Neste caso, a propriedade-* só se aplica aos demais sujeitos do sistema. Por exemplo, o conceito de sujeitos de confiança pode ser usado 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 é a dos subsistemas mais críticos do sistema operacional, como gerência de memória e *drivers* de dispositivos [Amoroso 1994].

Um dos principais problemas do modelo Bell-LaPadula reside no aspecto extremamente restritivo da propriedade-*.[9] 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. Este fenômeno é conhecido como **superclassificação da informação** [Landwehr 1981]. 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.

Modelo Biba

O modelo Bell-LaPadula tem por objetivo conter ameaças de revelação não-autorizada; não obstante, os próprios criadores do modelo BLP discutem como ele poderia ser adaptado para conter ameaças de integridade [Bell 1976]. Embora as idéias de Bell e LaPadula careçam de maior consistência, elas serviram de base para que Ken Biba desenvolvesse um modelo de segurança com o propósito de garantir a integridade da informação; conhecido como **modelo de integridade Biba** [Biba 1977].

O modelo *Biba* é definido como o dual do *BLP*. Suas regras são similares ao do modelo anterior, mas tem como objetivo a preservação da integridade das informações classificadas, evitando alterações não autorizadas. O modelo define níveis hierárquicos de integridade para os sujeitos (s_i's) e para os objetos (o_j's) similares aos níveis de sensitividade definidos no *BLP*. A **propriedade simples de integridade** define que um sujeito só pode ler um objeto se o seu nível de integridade for dominado pelo do objeto. A **propriedade estrela de integridade** especifica que um sujeito pode ter direito de escrita sobre um objeto, se e somente se o seu nível se sensibilidade for dominado pelo do objeto.

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

Existem alguns outros modelos mandatórios além do *BellLapadula* e do *Biba* citados em literatura. O modelo *Clark-Wilson* (*CW*) é um exemplo, baseia-se na idéia que a integridade é mais importante que a confidencialidade [Clark. e Wilson. 1987] em operações comerciais. Porém, 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

[9] Segundo Landwehr [30], a provisão de sujeitos de confiança é um reconhecimento de que a propriedade-* impõerestrições de acesso mais rigorosas do que aquelas usadas extracomputacionalmente em ambientes de segurança militar,uma vez que o seu propósito é evitar que programas malcomportados causem vazamentos de informação.

distinto na seqüência de atividades que formam uma transação) como essência de sua definição.

4.3.3. Modelos Baseados em Papéis (RBAC: Role Basic Access Control)

Os modelos baseados em papéis regulam o acesso dos usuários à informação com base nas atividades que os usuários executam no sistema. Estes modelos necessitam a identificação de **papéis** no sistema, onde um papel pode ser definido como um conjunto de atividades e responsabilidades associadas a um determinado cargo ou função. Logo, ao invés de especificar um conjunto de acessos autorizados para cada usuário do sistema, as permissões são conferidas aos papéis. Por conseguinte, um usuário que exerce um papel pode realizar todos os acessos para os quais o papel está autorizado.

Os modelos baseados em papéis possuem diversas características importantes, tais como: [Sandhu e Samarati (1994)]:

> **Gerência de autorizações mais simples** - a especificação de autorizações é dividida em duas partes, associação de direitos de acesso a papéis e associação de papéis a usuários. Isso simplifica bastante a gerência da segurança, facilitando tarefas como ajustar os direitos de acesso de um usuário em função de uma promoção ou transferência de setor na organização.
>
> **Suporte a hierarquias de papéis** - em muitas aplicações existe uma hierarquia natural de papéis baseada nas noções de generalização e especialização. Isto permite que permissões sejam herdadas e compartilhadas através da hierarquia.
>
> **Suporte a privilégio mínimo** - os papéis permitem que um usuário trabalhe com o mínimo privilégio exigido para uma determinada tarefa. Usuários autorizados a exercer papéis poderosos só precisam exercê-los quando forem absolutamente necessários, minimizando a possibilidade de danos por causa de erros inadvertidos.
>
> **Suporte a separação de tarefas** - os modelos baseados em papéis suportam separação de tarefas. Nestes modelos, a separação de tarefas é obtida através de restrições à autorização e/ou à ativação de papéis considerados mutuamente exclusivos.
>
> **Delegação da administração de segurança** - modelos baseados em papéis permitem que a administração da segurança seja descentralizada de maneira controlada. Isto significa que o administrador de segurança pode delegar parte de suas atribuições de acordo com a estrutura organizacional ou com a arquitetura do sistema computacional, permitindo, por exemplo, que administradores regionais gerenciem a segurança dos subsistemas locais.

O modelo RBAC é **independente de política**, diferente do que acontece com os modelos tradicionais de controle de acesso.[10] A independência da política possibilita uma grande flexibilidade e facilidade do ajuste do controle de acesso à medida em que ocorram mudanças no ambiente. Apesar da independência de política, o RBAC garante três princípios de segurança: o princípio de privilégio mínimo, separação de tarefas

[10] Tanto o controle obrigatório quanto o discricionário impõem uma política de segurança. No MAC, fluxos de informação contrários a um determinado sentido no reticulado de rótulos de segurança são proibidos; no DAC, a política imposta é que o dono do objeto é quem determina os seus direitos de acesso.

(restrito aos papéis) e abstração de dados (não há restrições quanto à natureza das permissões, podendo ser abstratas, tais como débito e crédito em um objeto conta).

O Modelo RBAC-NIST[11] [Sandhu e Park 2004]

O modelo RBAC-NIST reflete a compreensão e a modelagem do RBAC por parte de dois grupos de pesquisa: o grupo do NIST e o grupo liderado por Ravi Sandhu, da George Mason University. O modelo NIST-RBAC é um excelente componente para uma padronização do conhecimento na área de controle de acesso baseado em papéis.

Sendo o RBAC um conceito bastante amplo e aberto, bem como complexo de ser representado, a utilização de um modelo único para tratá-lo torna-se um tanto quanto restritiva e complexa. Uma abordagem mais realista seria a definição de uma família de modelos, que a partir de um modelo básico que contempla as características fundamentais do RBAC, modelos adicionais podem ser criados com mais funcionalidades e requisitos em relação ao básico.

O modelo RBAC-NIST é definido por quatro modelos:

Modelo RBAC Básico (Core)

O modelo RBAC básico define um conjunto de elementos e relações para ativar o sistema RBAC completamente. Isto inclui as relações usuário-papel e permissão-papel. O RBAC básico também introduz o conceito de ativação do papel como parte da sessão do usuário dentro do sistema computacional. Ele é necessário em qualquer sistema RBAC, mas os outros componentes são independentes entre si e podem ser implementados separadamente

Este modelo inclui os conjuntos de 5 elementos básicos de dados chamado usuários (pessoas, hosts, etc), papéis, objetos operações e permissões, e também as relações entre eles. A Figura 4.5 (RBAC básico) apresenta estes elementos junto com as suas relações.

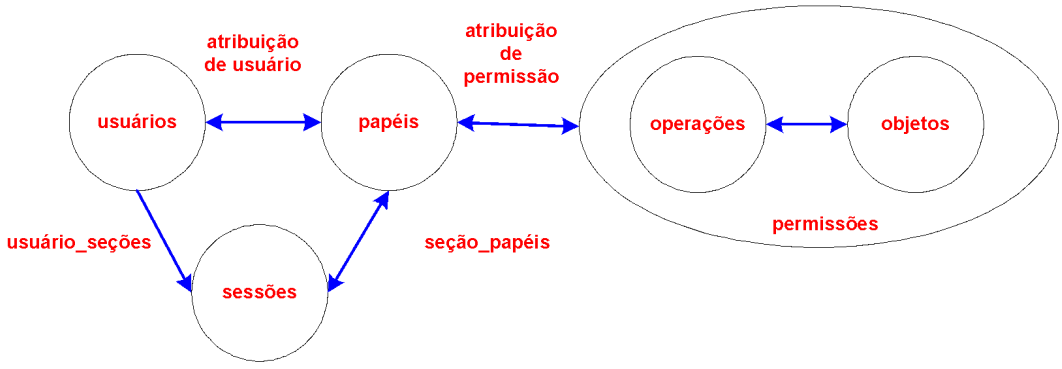

Figura 4.5. RBAC Básico (Core).

[11] National Institute of Standards and Technology

O modelo RBAC como um todo é definido basicamente pelos seus usuários relacionados a papéis (autoridade ou responsabilidade atribuída a um usuário) e as permissões (aprovação para executar uma operação em um ou mais objetos protegidos) relacionadas a estes papéis. Assim, um papel admite que sejam estabelecidos relacionamentos de muitos para muitos entre usuários e permissões. O modelo RBAC básico inclui um conjunto de seções onde cada uma delas é o mapeamento entre um usuário e o subconjunto de papéis a ele relacionado.

A Figura 4.5 mostra as relações da atribuição do usuário e a atribuição da permissão, onde as setas representam os relacionamentos de muitos para muitos (um usuário pode estar relacionado a um ou mais papéis e um papel pode estar relacionada a um ou mais usuários). Esta estrutura possibilita uma grande flexibilidade e possibilidades de atribuições de permissões a papéis e usuários, evitando que o usuário possa ter acesso a recursos desnecessários (o controle de acesso é limitado ao tipo de acesso que pode estar associado a usuários e recursos).

Cada sessão é um mapeamento de um usuário para alguns dos possíveis papéis que ele pode assumir durante um determinado período de tempo. Cada sessão está associada a um único usuário e cada usuário associado a uma ou mais sessões.

Modelo RBAC Hierárquico

O modelo RBAC hierárquico adiciona as relações (Figura 4.6), ao modelo básico, que suportam a hierarquia de papéis. As hierarquias são meios naturais de estruturar os papéis, representando os aspectos de autoridade e responsabilidade dentro de uma organização.A hierarquia é matematicamente uma ordem parcial definindo a relação de superioridade entre papéis, pelo qual os papéis superiores adquirem as permissões dos seus papéis subordinados e os papéis subordinados adquirem usuários dos seus papéis superiores.

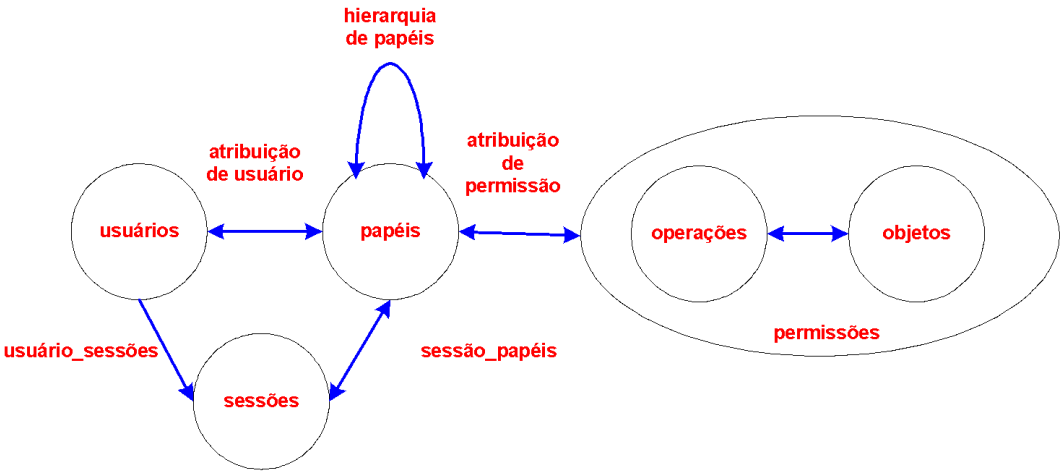

Figura 4.6. RBAC hierárquico.

Apesar das hierarquias arbitrárias estejam mais próximas de uma realidade, a larga utilização de hierarquias limitadas levou a uma subdivisão do RBAC Hierárquico:

- **RBAC Hierárquico Geral** - uma hierarquia de papéis pode constituir qualquer tipo de ordem parcial existente. suporte para uma determinada ordem parcial arbitrária que serve como uma hierarquia de papéis, para incluir o conceito de múltiplas heranças de permissões e usuários entre os papéis existentes
- **RBAC Hierárquico Limitado** - quando existe qualquer restrição em relação à estrutura da hierarquia de papéis. Na maioria das vezes, as hierarquias estão limitadas a estruturas simples como árvores ou árvores invertidas.

A hierarquia de papéis define a relação de herança entre os papéis. A herança é descrita em termos de permissões, de forma que: r1 herda o papel de r2 se todos os privilégios de r2 são também privilégios de r1.

O padrão NIST reconhece tanto as hierarquias geral e limitada, como apresentado anteriormente. A Figura 4.7 apresenta um exemplo de uma estrutura hierárquica de papéis. Pode-se observar que os usuários no topo da árvore além de possuírem as suas permissões, eles herdam as permissões dos usuários que estão abaixo dele (o diretor possui as suas permissões mais as do chefe de Depto. e Encarregado de Divisão).

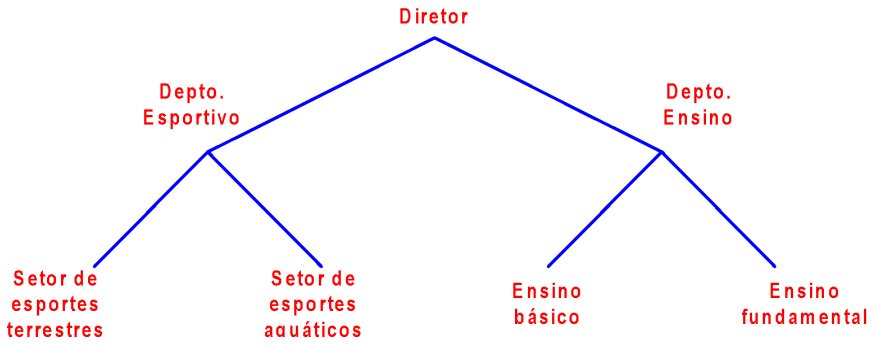

Figura 4.7. Estrutura hierárquica de papéis.

Modelo RBAC com Restrições

A separação de tarefas é utilizada para aplicar políticas de conflito de interesses, de forma que estas políticas previnam que usuários excedam a sua autoridade em suas posições de trabalho.

O princípio de separação de tarefas tem como objetivo garantir que as falhas por omissão e delegação de poderes dentro de uma organização possam ser causados somente como resultado da conivência entre indivíduos. Para minimizar a possibilidade destas conivências, indivíduos de diferentes habilidades ou conhecimento profissional ou interesses são atribuídos a diferentes tarefas necessárias no desempenho da função no negócio da Instituição. A motivação é garantir que a fraude e maiores erros não venham ocorrer sem conivência deliberada de vários usuários. Dois modelos são apresentados a seguir:

Separação Estática de Tarefas (SET) – Static Separation Duty (SSD)

O modelo de separação estática de tarefas adiciona relações exclusivas entre os papéis com respeito às atribuições do usuário (Figura 4.8). Isto significa que ele aplica restrições de associações de usuários a funções, de forma que quando um usuário está associado a um papel, ele não poderá assumir outro. Por exemplo, se um usuário exerce o papel de comparador, ele também não pode exercer o papel de executor de pagamentos (confeccionar os cheques). Este tipo de procedimento evita possíveis fraudes, tornando estes papéis mutuamente exclusivos. Normalmente, as restrições estáticas são colocadas em operações administrativas que possuem um potencial para questionar as políticas de separação de tarefas nos altos escalões administrativos.

Os modelos RBAC definem relações de separação estática de serviços com respeito a restrições nas associações usuário-papel (ex: um usuário só pode estar associado a um único papel por vez). Esta definição é bastante restritiva em dois aspectos importantes: o tamanho do conjunto de papéis e a combinação de papéis no conjunto para que a atribuição do usuário é restrita. Assim, o modelo define a separação estática de tarefas em dois argumentos: o conjunto de papéis e a maior cardinalidade que indica uma violação da separação estática de tarefas (ex: uma organização prescreve que nenhum usuário do setor de compras pode estar associado a três papéis dos quatro existentes). Porém, deve-se ter cuidado que a herança de usuários não questione as políticas de separação de tarefas.

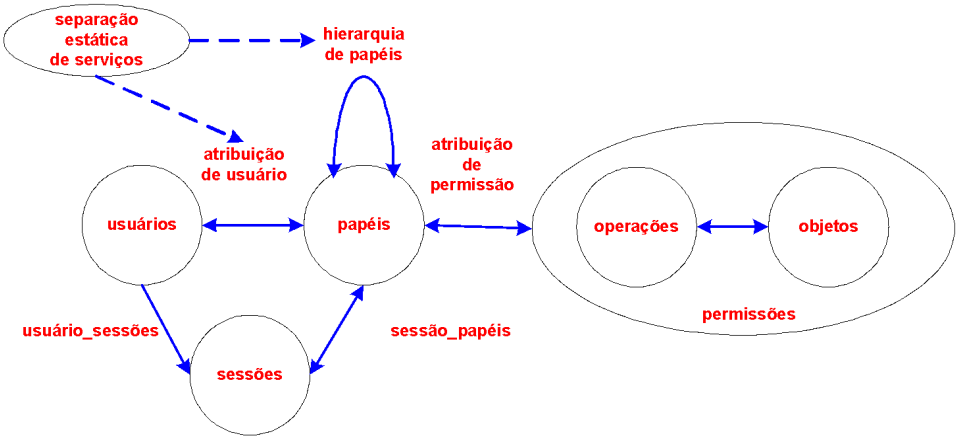

Figura 4.8. Separação estática de tarefas.

Separação Dinâmica de Tarefas (SDT) – Dynamic Separation Duty (DSD)

A separação dinâmica de tarefas define relações exclusivas com respeito a papéis que são ativados como parte da sessão do usuário (Figura 4.9).

A separação dinâmica de tarefas coloca as restrições somente nos papéis que podem ser ativados dentro ou via as sessões dos usuários, enquanto a estática coloca as restrições em todo o espaço de permissões do usuário.

A separação dinâmica de tarefas possui a capacidade de encaminhar as questões de conflito de interesses no momento que o usuário é associado a um papel. Esta política admite que um usuário seja autorizado a exercer dois ou mais papéis que não criem

conflitos de interesse quando atuando de forma independente, mas deve atender aos interesses da política quando ativados simultaneamente. Por exemplo: um usuário pode ser autorizado para exercer as funções de caixa e de supervisor de caixa, onde o supervisor é utilizado para reconhecer as correções do dinheiro da gaveta aberta do caixa. Caso a pessoa esteja exercendo somente o papel de caixa, ao mudar para o papel de supervisor de caixa, esta deverá inicialmente fechar o seu caixa antes de assumir o novo papel (supervisor de caixa).

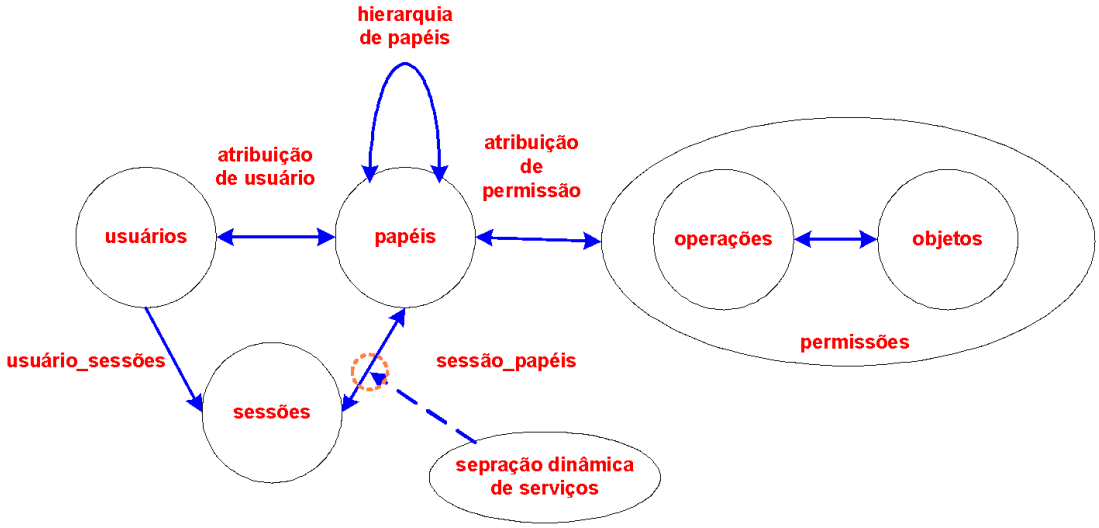

Figura 4.9. A separação dinâmica de tarefas.

Sendo assim, viu-se que o RBAC é um modelo com características diferenciadas do DAC e MAC, que faz uso do papel como aspecto principal de sua abordagem, e que possibilita uma fácil adaptação de novos requisitos em políticas de controle de acesso centralizadas de forma bastante flexível.

4.4 Digital Rights Mangement - DRM

Os direitos digitais são hoje um alvo de diversas atividades como: permissão para reprodução, transferência ou empréstimo de arquivos digitais, impressão, uso, extração e edição de informações disponibilizadas de forma digital, bem como a inserção e a obtenção de cópias de publicações digitais. No ambiente virtual, a possibilidade de gerenciamento e o controle e segurança na proteção do bem é um dos maiores problemas para os seus produtores. Os dispositivos tecnológicos buscam o controle e objetivam minimizar a disseminação ou a distribuição não autorizadas destes produtos, garantindo os direitos autorais de seus donos (ex: copiar o trabalho, emitir cópias do trabalho para o público, executar apresentações em público). [Kaminsky 2004]

A gerência dos direitos autorais (DRM - Digital Right Management) busca minimizar estes problemas, apresentando soluções compatíveis com as solicitações dos proprietários de bens, e busca via recursos tecnológicos disponíveis, com base numa política de controle de acesso a softwares, músicas, filmes ou outros dados digitais

restringir o uso destes dispositivos atendendo os interesses de direitos de cópia de seus proprietários. [Kaminsky 2004]

A necessidade de se utilizar o DRM está ligada às diferenças existentes entre o mundo digital e não digital levantando questões como: [Duncan, Barker, Douglas, Morrey e Waelde 2004 e Bechtold 2001]:

- O pronto acesso aos recursos na Internet cria dificuldades em estabelecer a fonte destes, enquanto é mais fácil incluir uma declaração de direitos de cópia em uma mídia de papel;
- A tranqüilidade com que os recursos digitais podem ser modificados encoraja a modificação sem verificar se esta é permitida (em mídias como papel esta questão está limitada a cópias com alterações de pequenos pedaços); e
- Qualquer pessoa pode publicar material na Internet sem as devidas permissões. O monitoramento destes aspectos é complicado, porque a maioria das pessoas não é capaz de definir legalmente a licença de publicação (em mídias como papel este controle é mais efetivo).

4.4.1 Aspectos Gerais do funcionamento de um Sistema DRM

Um típico sistema DRM leva em consideração basicamente três componentes na sua configuração [Ku e Chi 2004]:

- Proprietário do conteúdo – normalmente possui todos os direitos do conteúdo;
- Gerente – manipula todas as transações em nome do proprietário do conteúdo, trata as questões da licença especificando exatamente as permissões para um usuário fazer uso do conteúdo; e
- Usuário – neste caso, refere-se ao hardware ou software confiável, servindo de proxy para o consumidor do produto. Este hardware ou software é considerado confiável porque não admite que consumidores não autorizados acessem o conteúdo.

A Figura 4.10 apresenta uma visão geral de um típico sistema DRM.

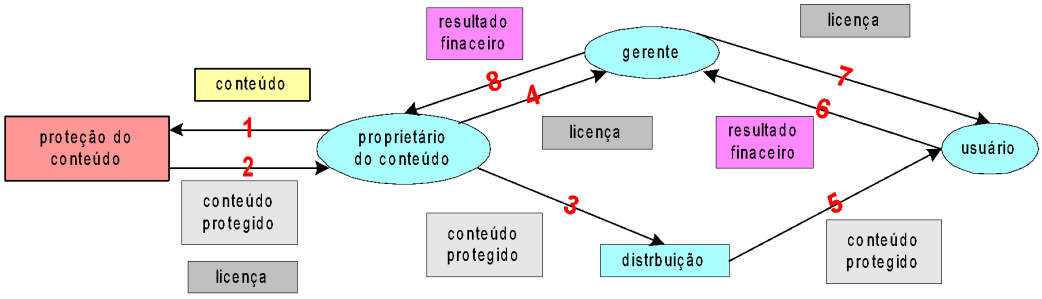

Figura 4.10. Sistema DRM.

1 – o proprietário entra com o conteúdo para ser protegido. Em algumas situações o conteúdo pode ser codificado em algum determinado formato. Por exemplo, o sistema DRM da Microsoft requer o formato Window Media (.wma ou .wmv). O proprietário do conteúdo pode desejar inserir uma maca d'água digital no conteúdo com o propósito de identificação. O sistema DRM pode então criptografar (na maioria das vezes fazendo

uso de técnicas de criptografia proprietária) e empacotar para a distribuição. O proprietário do conteúdo necessita especificar, utilizando uma Linguagem de Expressão de Direitos (REL), todos os direitos ou regras de uso que se aplicam ao conteúdo. Pode ser necessário que as regras sejam divididas em conjuntos, no qual cada conjunto está ligado a um determinado contexto, apesar de ser um mesmo conteúdo.

2 – o sistema DRM retorna o conteúdo protegido e a licença (ou um conjunto de licenças). A licença contém todos os direitos aplicáveis, termos e condições para o uso do conteúdo. Ele também contém a chave que é necessária para decriptografar o conteúdo protegido.

3 – o proprietário do conteúdo dissemina o conteúdo protegido via vários canais de distribuição, como a Internet, CDs, DVDs, email, P2P filesharing, entre outros. Os dois últimos meios de distribuição formam o conceito de superdistribuição. Este conceito se refere a possibilidade dos usuários redistribuir o conteúdo protegido de forma livre sem nenhuma restrição.

4 – o proprietário do conteúdo envia a licença/licenças para o gerente. O gerente é uma entidade confiável que pode manipular todas as solicitações de transações para acessar o conteúdo. Ele libera alguns recursos e possibilita que o proprietário do conteúdo se concentre no desenvolvimento do conteúdo, como também pode realimentar o perfil de consumo do usuário.

5 – o usuário recupera o conteúdo protegido do canal de distribuição. Ele examina o seu meta-data para identificar a licença necessária para acessar o conteúdo e a localização do gerente, que irá prover a licença.

6 – caso o usuário (consumidor) não possui a licença ou não é válida, ele pode contactar o gerente para solicitar uma licença e realizar o pagamento necessário.

7 – após o usuário realizar o pagamento, o gerente pode emitir a licença. O tipo de pagamento realizado determina os direitos de acesso ao conteúdo.

8 – o gerente remete ao proprietário do conteúdo o resultado financeiro das transações (após a dedução do seu serviço). Ele também pode prover alguma informação proveitosa de cada transação.

4.4.2 Necessidades do Sistema DRM

Implementação de hardware e software

O DRM é inicialmente disponibilizado somente em PC. Sendo o PC o sistema do usuário final, quando conectado a Internet, torna-se uma ferramenta de fácil liberação de conteúdo, realização de atualizações e *downloads* de software de segurança e DRM. O PC é considerado uma implementação de software.

A proteção do conteúdo também pode vir na forma de implementação de hardware. Por exemplo, o contorno da infraestrutura para a área de armazenamento de dados em discos óticos, protegido contra acessos não autorizados. Esta área só pode ser acessada por hardware em concordância com o acesso.

Interoperabilidade e Mobilidade

A Internet é quase sempre acessível para todas as formas de elementos computacionais, onde a natureza heterogênea do cenário computacional não limita o acesso à Internet. O DRM poderia ter a mesma acessibilidade, mas eles, na sua maioria, são sistemas proprietários empregando formato de dados proprietários e técnicas de criptografia confiáveis. Estes aspectos proporcionam uma grande ausência de interoperabilidade entre sistemas DRM diferentes, limitando os usuários no seu uso.

Pode-se ainda citar que como a maioria das licenças DRM está limitada a dispositivos e não a usuários, obtendo o acesso ao conteúdo somente ao dispositivo liberado.

Segurança

A segurança é um requisito fundamental no DRM. Necessidades essenciais de segurança nos sistemas DRM incluem: confidencialidade e integridade do conteúdo (obtida via o uso de criptografia, assinaturas e certificados digitais), identificação única do usuário para o controle de acesso (pode ser verificado no agente) e mecanismos de proteção à falsificação (preocupação com duas importantes áreas: o conteúdo protegido e o player do usuário final) para processar o conteúdo protegido e aplicar as regras de uso do conteúdo.

Na maioria dos sistemas DRM existentes, a preocupação com a segurança é com o conteúdo liberado (que pode prover confidencialidade e integridade) no canal ao invés do conteúdo propriamente dito. Logo, este tipo de decisão é um aspecto de fraqueza na segurança, porque não evita a cópia e a redistribuição ilimitada do conteúdo protegido. Com isso, a proteção do conteúdo tem que ser persistente, evitando que estes tipos de ações possam ser realizados.

Privacidade do Usuário

Os usuários desejam opções para consumir o conteúdo de forma anônima e não ter o seu comportamento de consumo estabelecido em um perfil. O sistema DRM deve estabelecer parâmetros para que este tipo de exigência de seus usuários seja cumprida.

4.4.3 Componentes do Sistema DRM

Apresentada as funcionalidades e as necessidades de um sistema DRM, necessita-se conhecer os componentes de um sistema DRM. Esta seção visa apresentar os componentes básicos do DRM. A Figura 4.11 apresenta os principais componentes dos sistemas DRM, onde a proteção do conteúdo é representada por uma caixa que pode ser visualizada da seguinte forma [Ku e Chi 2004]:

1 – o conteúdo é rotulado com um identificador único, junto com o meta-dada (dado sobre o dado);

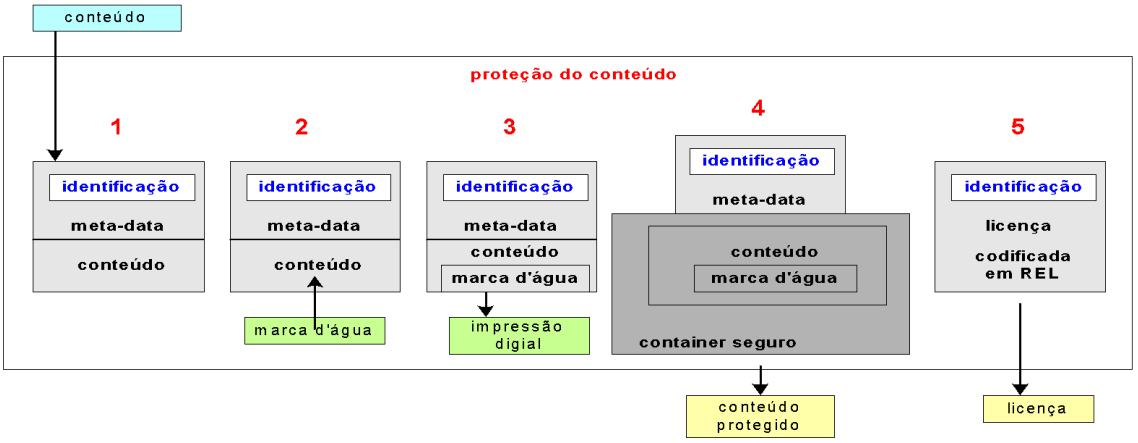

Figura 4.11. Componentes do sistema DRM.

2 – a marca d'água digital é inserida dentro do conteúdo para servir como uma prova de identidade do dono no evento de uma disputa.

3 – a impressão digital é gerado do conteúdo. É utilizado na aplicação para a autenticação, como identificação automática do conteúdo.

4 – o conteúdo está cercado por um container seguro, prevenindo acessos não autorizados.

5 – a licença, os direitos e as condições do uso do conteúdo são codificados em uma REL.

4.4.3.1 Identificação do Conteúdo e Meta-data

Antes dos direitos do conteúdo estarem definidos, ele tem de ser identificado para que os usuários que desejam acessa-lo possam comprar os seus direitos de uso. O meta-data do conteúdo pode prover alguma informação não sensível, como tipo de mídia, tamanho do arquivo, etc. Ele pode descrever alguma informação de como fazer uso do identificador do conteúdo.

Identificação do conteúdo

O identificador do conteúdo deve ser único para ser persistente. Apesar de mudanças no conteúdo, o identificador deve ser mantido. São utilizados esquemas de numeração padrão no DRM como: ISBN, ISSN, ISAN e DOI (Digital Object Identifier). A ACM utiliza o sistema DOI para itemizar cópias digitais de vários *proceedings* na sua biblioteca digital [DOI 2004].

O identificador do conteúdo também pode ser utilizado para localizar recursos ou derivações do conteúdo. Isto pode ser útil por exemplo, quando o usuário está interessado em localizar a versão do conteúdo para ser utilizado em seu dispositivo.

Meta-data

O meta-data complementa o uso do identificador do conteúdo (string alfanumérica). Se a infraestrutura do identificador do conteúdo (por exemplo DOI) é conhecido, então o identificador do conteúdo pode agir como um ponteiro para mais informações. Por outro

lado, o meta-data atenderia a esta capacidade e o contexto do DRM poderia prover mais informação para acessar o conteúdo.

4.4.3.2 Identificação/Autenticação do Usuário

Esta questão é importante porque deseja-se que somente usuários autorizados sejam habilitados a acessar o conteúdo. O problema está na dificuldade de identificar o usuário na maioria dos sistemas DRM, porque o conteúdo está relacionado ao dispositivo ao invés do usuário. Por exemplo, o DRM Microsoft admite relacionar o conteúdo de áudio exatamente a uma máquina via o uso do número serial do hardware como entrada. Isto quer dizer que o usuário não pode acessar facilmente o conteúdo que ele pagou via o uso de seus dispositivos (ex: PC caseiro ou do seu trabalho), ele está autorizado a acessar somente de um dispositivo designado.

A identificação/autenticação do usuário pode ser delegada ao gerente que pode utilizar tecnologias como SSL para superar este problema. Existe também o conceito de *Single Sign-On* (SSO) pelo qual os usuários somente fazem *login* para acessar os serviços via múltiplas plataformas. O Microsoft DRM necessita que os usuários DRM registrem antes o seu serviço SSO para poder acessar o conteúdo.

4.4.3.3 Marca D'água Digital

A tecnologia de marca d'água pode ser utilizada para controle de cópia, identificação de conteúdo e cópia. A maioria das técnicas de marca d'água usa uma abordagem que é a inserção de um sinal ruidoso com pequena intensidade dentro do conteúdo. Esta marca d'água pode ser detectada via a utilização de métodos específicos e relacionados com a chave secreta, responsável em detectar e remover a marca d'água pelas partes autorizadas [Katzenbeisser e Veith 2003].

No DRM, o conteúdo é tipicamente vulnerável a ataques nos sistemas dos usuários finais. O conteúdo pode ser capturado durante a sua reprodução ou ter os seus mecanismos de proteção removidos pelos ataques diretos. A marca d'água pode ser utilizada para detectar cópias ilegais de conteúdo que estão desprotegidos quanto a este tipo de ataques. Esta detecção é realizada pelos sistemas de usuários finais que detectam a marca d'água e a ausência de um mecanismo de proteção associado que é suposto para vir com o conteúdo. O sistema do usuário final pode também reportar e assistir na trilha destas cópias ilegais.

As necessidades básicas da marca d'água são:

Imperceptível – a marca d'água não deve afetar a qualidade do conteúdo;

Segurança – a marca d'água deve ser somente acessível pelos parceiros autorizados; e

Robustez – a marca d'água deve ser persistente e resistente a ataques.

4.4.3.4 Identificação baseada no Conteúdo (Impressão digital)

A identificação baseada no conteúdo usa de características existentes no conteúdo com base nas suas representações (sinais e características) e as compara com entradas existentes no banco de dados. O termo impressão digital tem sido utilizado em conjunto

com a marca d'água. A impressão digital é diferente da marca d'água, sendo vista na seguinte comparação (Tabela 4.2):

Tabela 4.2. Tabela comparativa entre marca d`água e impressão digital.

Marca d'água	Impressão digital
Embute o sinal no conteúdo, alterando-o	Não embute o sinal no conteúdo.
Não é em função do conteúdo.	É em função do conteúdo.
Requer acesso prioritário ao conteúdo.	Não requer esta prioridade e pode ser usado para a legalidade do conteúdo.
Deve ser refeito para todas as cópias no caso de novas tecnologias.	Não possui esta necessidade.
Nenhum tratamento adicional para novos conteúdos.	Existe a necessidade de armazenar as impressões digitais dos novos conteúdos em um banco de dados.

4.4.3.5 Containers Seguros

Os containers seguros são implementados com algoritmos de criptografia tais como DES (Data Encryption Standart) e AES (Advanced Encryption Standart) [Buenett e Paine 2002]. Junto com certificados e assinaturas digitais o container seguro oferece um conteúdo com confidencialidade e integridade. A integridade pode ir mais adiante com a utilização de mecanismos de autenticação para o conteúdo.

Um aspecto interessante é que o uso do conceito de container seguro foge ao conceito da transação comercial eletrônica tradicional, pelo qual a chave do conteúdo e o conteúdo protegido são transmitidos juntos na mesma transação. No DRM, a proteção do conteúdo é realizada offline e a chave do conteúdo é obtida separadamente. A mesma chave pode ser utilizada para várias transações (distribuição do conteúdo protegido), tornando-se uma vulnerabilidade.

As regras de uso do conteúdo podem ser codificado em seu meta-data ou em licenças. Codificando nas licenças, possibilita uma maior flexibilidade na determinação específica das regras de uso, porque as regras irão basear-se nas necessidades do usuário.

4.5. The $UCON_{ABC}$ Usage Control Model

O UCON (Usage Control) é um novo modelo de controle de acesso, diferente dos modelos tradicionais, em que a autorização pode ser feita também em tempo de requisição. O UCON é um modelo que estende os modelos de controle de acesso tradicionais em vários aspectos. O acesso pode ser uma ação instantânea, ou pode ser uma ação contínua, durante um determinado período de tempo, com várias ações seqüenciais e próximas. A decisão de acesso pode ser realizada antes, durante o processo de acesso, ou em ambos os casos, e as ações durante o período de acesso podem resultar em alterações de atributos [Sandhu e Park 2004]. Esta seção irá apresentar as características básicas deste modelo, mostrando o quanto ele modifica alguns conceitos de controle de acesso já conhecidos e o quanto ele melhora o controle propriamente dito.

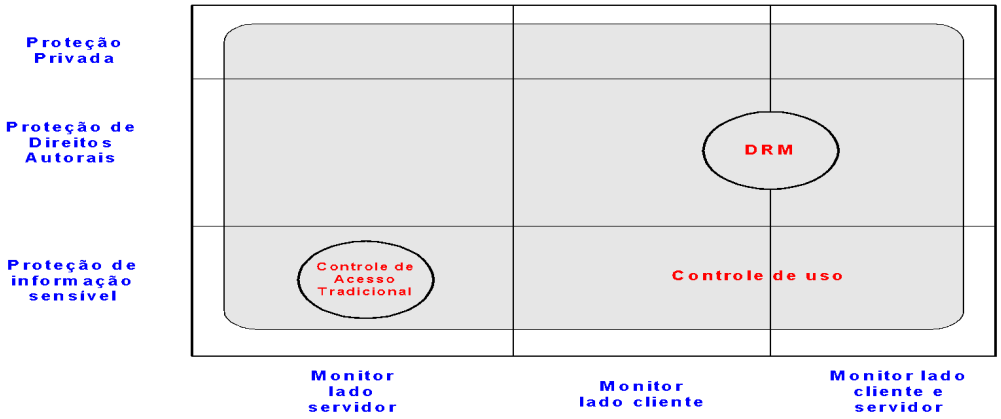

Figura 4.12. Cobertura do modelo UCON.

As decisões de uso no UCON são realizadas por políticas de autorização, obrigação e condição - UCONABC.

> As decisões de autorização são determinadas por políticas, utilizando atributos do sujeito, objeto e direitos.
>
> As obrigações são ações que tem que ser executadas por sujeitos, antes ou durante, o processo de acesso.
>
> As condições são necessidades pertinentes ao sistema ou ao ambiente em que opera que tem que ser satisfeitas antes ou durante o acesso.

A Figura 4.12 mostra a cobertura do UCON sobre os controles de acesso e seus relacionamentos com outros modelos.

4.5.1 Aspectos gerais do UCON

No UCON, os objetos estão relacionados com consumidores, provedores e identificadores. O consumidor busca o acesso ao objeto oferecido pelo provedor. O objeto pode conter informações privadas de elementos. Estes elementos são chamados de identificadores e guardam certos direitos no objeto. A decisão de uso é baseada nos relacionamentos entre estes diferentes componentes (consumidor, provedor e identificador) em relação ao objeto, e não mais tomada numa só direção. A parte do núcleo do UCON negocia com os aspectos relativos à tomada de decisão em relação da utilização dos objetos pelos consumidores.

Tradicionalmente, o controle de acesso tem negociado com autorizações como a base para o seu processo de tomada de decisão. No modelo UCON_{ABC}, o processo de tomada de decisão utiliza os atributos do objeto e do sujeito. Os atributos podem ser: identidades, rótulos de segurança, propriedades, capacidades, etc. Os predicados de segurança, obrigações e condições, podem ser avaliados antes ou durante o exercício de uma requisição. Em adição, o uso de um objeto pode necessitar de atualizações nos atributos do usuário (sujeito) ou do objeto antes, durante ou após o exercício do uso do objeto/recurso.

As seguintes propriedades distinguem o UCON dos modelos de controle de acesso tradicionais [Sandhu e Park 2004]:

A continuidade do processo de decisão de acesso; e

A mutabilidade dos atributos do sujeito e do objeto.

A continuidade e mutabilidade no UCON introduzem os conceitos de interatividade e concorrência, onde o acesso resulta na atualização dos atributos do sujeito ou do objeto. Estas mudanças, por outro lado, resultarão alterações durante ou em futuros acessos pelo mesmo sujeito, objeto ou algum acesso que esteja implicitamente relacionado. Logo, estas mudanças podem alterar não só o estado do acesso, mas também dos que estiverem relacionados.

4.5.2 Componentes do modelo $UCON_{ABC}$ [Sandhu e Park 2004]

O modelo $UCON_{ABC}$ possui oito componentes: sujeitos (usuários), atributos dos sujeitos objetos (recursos), atributos dos objetos, direitos, autorizações, obrigações e condições. As autorizações, obrigações e condições são predicados funcionais que existem para serem avaliados em uma decisão de uso. Os sujeitos, objetos, e direitos podem ser divididos em vários componentes, detalhados com diferentes perspectivas.

O controle de acesso tradicional utiliza somente autorizações para o processo de decisão. As obrigações e condições são os novos conceitos, os quais podem resolver certas deficiências encontradas em modelos de controle de acesso tradicionais. Outro aspecto significante do $UCON_{ABC}$ é que os atributos do sujeito e do objeto podem ser mutáveis, isto é, são mudados como conseqüência do acesso. Por exemplo: políticas que exigem limites no número de acessos pelos usuários, podem ser facilmente especificadas usando atributos mutáveis.

A Figura 4.13 apresenta os relacionamentos entre os componentes. O processo de decisão é mostrado como o relacionamento entre sujeitos, objetos e direitos que necessitam autorizações, obrigações e condições.

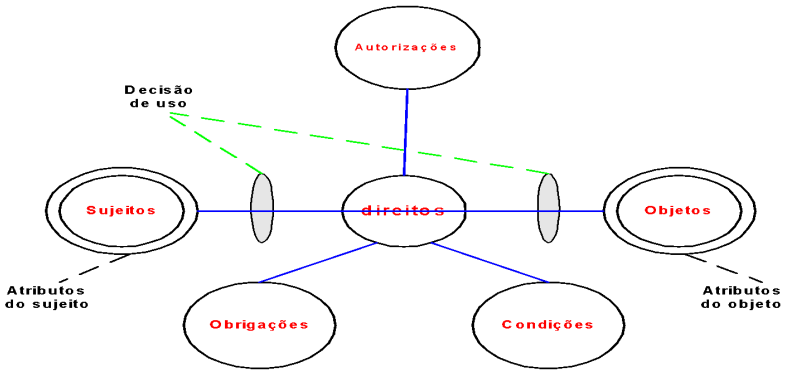

Figura 4.13. .O relacionamento dos componentes UCON.

Sujeitos e seus atributos

O sujeito (usuário) é uma entidade com atributos e determinados direitos de execução nos objetos (recursos). Os atributos do sujeito são propriedades ou capacidades que podem ser utilizadas no processo de decisão do uso. Exemplos de atributos: identidades, nomes de grupos (conjunto de usuários que possuem os mesmos direitos), papéis, etc. Os atributos no sujeito podem ser: [Sandhu, Park e Zhang 2004]

Imutável - ele não pode ser mudado pela atividade do usuário, somente ações administrativas podem mudá-lo; e

Mutável - pode ser modificado como efeito do acesso do usuário ao objeto/recurso (ex: atributo mutável - crédito).

Objetos e seus Atributos

Os objetos são basicamente os recursos oferecidos pelo sistema. Os objetos são também associados com atributos, e estes possuem certas propriedades que podem ser utilizadas nas tomadas de decisão. No caso do objeto classe, por exemplo, ele pode ser usado para estabelecer categorias de objetos, possibilitando que a autorização possa ser realizada não somente em um objeto, mas no conjunto de objetos que pertençam àquela classe. Alguns exemplos de atributos para objetos podem ser citados: valor, permissão, função, etc. Imagine a seguinte aplicação do atributo valor: o livro "A Volta dos que não Foram" necessita de R$ 100,00 para ler e mais R$ 50,00 para imprimir. Conforme o pagamento realizado, o sujeito poderá ler e ou imprimir o livro.

Direitos

São privilégios que um sujeito pode manter e exercer em um objeto. Os direitos consistem em um conjunto de funções de uso que habilita o acesso de um sujeito a objetos. Os direitos podem ou não ter hierarquia e podem ser divididos em: direitos do consumidor, direitos do provedor e direitos do identificador.

No UCON_{ABC}, o conceito de direito é semelhante ao do controle de acesso (direito de leitura e escrita, por exemplo). Porém, existe uma diferença: o UCON_{ABC} não visualiza os direitos como na matriz de acesso, independente da atividade do sujeito, eles estão relacionados com os atributos do sujeito, atributos do objeto, autorizações, obrigações e condições (ex. de direitos – uso de objetos, delegação de direitos e direitos para administração de acesso, etc). Imagine no caso do direito de ler um livro pela Internet, esse direito foi dado a:

Um determinado sujeito – atributo do sujeito – identificação;

Atributo do objeto livro – leitura e tempo indeterminado;

Obrigação – pagamento do valor estipulado; e

Condição – a leitura seja realizada somente no ambiente Windows.

Autorizações

As autorizações são predicados funcionais utilizados na avaliação da decisão de uso. As autorizações avaliam os atributos do sujeito, atributos do objeto e direitos requisitados junto com o conjunto de regras de autorização para a decisão de uso.

As autorizações podem ser:

Pré-autorização – é executada antes de um direito requisitado seja exercido.

Autorização em andamento – é executada enquanto o direito é exercido. A autorização em andamento pode ser executada continuamente ou periodicamente durante o tempo de acesso.

De forma geral, as políticas de controle de acesso tradicionais, incluindo MAC, DAC e RBAC utilizam, de alguma forma, a pré-autorização para as suas decisões, bem como o DRM em alguns casos. Algumas autorizações podem necessitar atualizações nos atributos dos objetos e sujeitos. Estas atualizações podem ser feitas antes, durante ou após (ex: créditos pré-pagos para utilização de recursos).

Obrigações

As obrigações são predicados funcionais que verificam as necessidades mandatórias que um sujeito tem que desempenhar antes ou durante o exercício do uso. As obrigações podem ser:

> Pré-obrigação – é um predicado que utiliza algum tipo de histórico de funções para verificar se certas atividades tenham sido realizadas ou não e retorna verdadeiro ou falso (ex: um usuário deve ter preenchido alguns dados antes de ler um documento de uma Empresa).
>
> Obrigação em andamento – é um predicado que tem que ser satisfeito continuamente ou periodicamente enquanto os direitos admitidos estão em uso (ex: um usuário tem que verificar certos avisos enquanto "logado").

As obrigações podem ou não possuir atributos. Os atributos podem ser utilizados para determinar quais tipos de obrigações são necessárias para a aprovação do uso. Pode-se dizer que os atributos não são usados para tomadas de decisão, com respeito a obrigações, mas são utilizados somente na escolha do que as obrigações aplicam-se.

Condições

São fatores de decisão com base no sistema ou no ambiente. O predicado condição avalia o status do sistema ou ambiente para verificar se as necessidades são satisfeitas ou não (*true* ou *false*).

Os atributos de objetos e sujeitos podem ser usados para escolher que condições têm que ser usadas para uma requisição. Portanto, nenhum atributo é incluído dentro de suas próprias requisições. Diferente das obrigações e autorizações, as variáveis das condições não podem ser mutáveis, porque as condições não estão diretamente sob o controle de sujeitos (ex: status de segurança de um sistema, carga do sistema, variação de fuso horário para transações comerciais).

4.5.3 Os principais modelos da família $UCON_{ABC}$ [Sandhu e Park 2004]

Apresentados os conceitos básicos do modelo UCON, esta seção visa apresentar o conjunto de modelos da família UCON. Os modelos que serão apresentados são considerados os principais, porque visam o processo de aplicação, não incluindo construtores administrativos. A classificação dos modelos é baseada nos seguintes critérios:

> Fatores de decisão – que consiste de autorizações, obrigações e condições;
>
> Continuidade de decisão – ou pré ou em andamento, com respeito ao acesso em questão; e

Mutabilidade – que pode admitir atualizações nos atributos dos sujeitos e objetos.

Se todos os atributos são imutáveis, nenhuma atualização é possível no processo de decisão. Este caso é denotado como '0'. Com atributos mutáveis, atualizações são possíveis antes, durante ou após o direito ser exercido. Denota-se 1, 2 e 3 respectivamente (Tabela 4.3).

Tabela 4.3. Os 16 modelos $UCOM_{ABC}$ básicos.

	0 (immutable)	1 (pre-update)	2 (ongoing-update)	3 (post-update)
preA	S	S	N	S
onA	S	S	S	S
preB	S	S	N	S
onB	S	S	S	S
preC	S	N	N	N
onC	S	N	N	N

Exemplo: Suponha que Alice é membro de uma biblioteca musical, e que ela pague R$ 1,00 por hora de música tocada. Este exemplo pode ser tratado como uma pré-autorização com atualização posterior, não existindo a necessidade de realizar alguma atualização durante a execução da música. Se o fator de decisão é durante, a atualização pode ocorrer antes, durante e após a execução. Explica as quatro primeiras linhas da tabela. Para as duas linhas restantes, o fator de decisão é a condição.

A Figura 4.14 mostra a continuidade de decisões, com as possibilidades de mutação de atributos que podem ocorrer antes ou durante a execução.

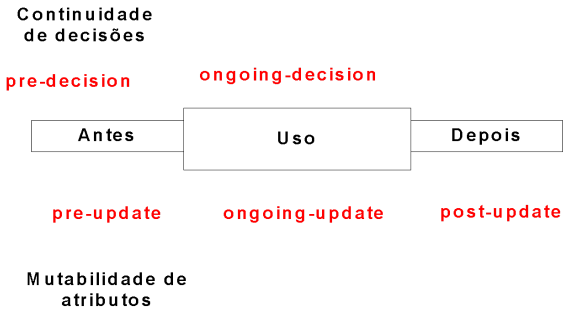

Figura 4.14. Continuidade de decisões.

A Figura 4.15 apresenta as possíveis combinações do modelo $UCON_{ABC}$ e suas relações, considerando-se que cada A, B e C estão na base do modelo. O próximo nível possui as combinações de dois deles e assim por diante. Desta forma, foi apresentada de forma sucinta que combinações de A, B e C podem ser utilizadas em um contexto.

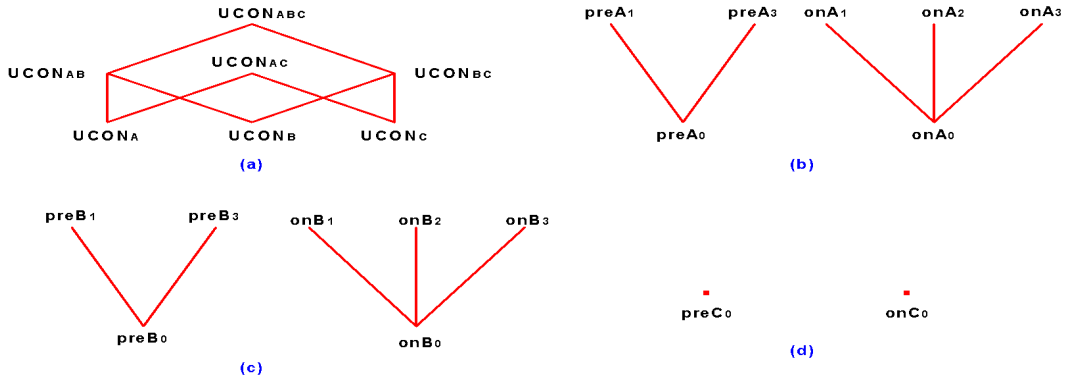

Figura 4.15. Combinações do modelo UCON.

A seguir serão apresentados alguns dos 16 modelos da família UCON. Esta apresentação visa mostrar a validação destes modelos como exemplificá-los.

Modelo de pré-autorização - $\text{UCON}_{\text{preA}}$

As autorizações têm sido consideradas como o núcleo do controle de acesso, como o uso da pré-autorização na tomada de decisão. O modelo $\text{UCON}_{\text{preA}}$ utiliza a pré-autorização no seu processo de decisão de uso, onde o processo de decisão de autorização é realizado antes do uso ser admitido. Existem três modelos detalhados baseados nas variações de mutabilidade.

> $\text{UCON}_{\text{preA0}}$ – modelo de pré-autorização imutável que não requer atualização. Ele possui os seguintes componentes.

S;O;R;ATT(S);ATT(O) and preA (subjects, objects, rights, subject attributes, object attributes, and pre-authorizations respectively);

> allowed(s; o; r) => preA(ATT(s);ATT(o); r).
>
> $\text{UCON}_{\text{preA1}}$ – é um modelo com procedimento de pré atualização opcional. Ele inclui funções de atualização que modificam os atributos antes do seu uso ser iniciado. O modelo é idêntico ao $\text{UCON}_{\text{preA0}}$ exceto pela adição dos seguintes processos de pré-atualização.

preUpdate(ATT(s)); preUpdate(ATT(o)), um procedimento opcional para executar operações de atualização em ATT(s) e ATT(o), respectivamente. Veja que o preUpdate pode incluir operações não determinísticas.

> $\text{UCON}_{\text{preA3}}$ - é um modelo com procedimento de pós-atualização opcional. A atualização posterior utiliza funções para modificar certos atributos após o uso ter encerrado. O modelo é idêntico ao $\text{UCON}_{\text{preA0}}$ exceto pela adição dos seguintes processos de pós-atualização.

postUpdate(ATT(s)); postUpdate(ATT(o)), um procedimento opcional para executar operações de atualização em ATT(s) e ATT(o), respectivamente. Veja que o preUpdate pode incluir operações não determinísticas.

Exemplo 1 MAC policies, UCONpreA0 :

> L é uma lattice de rótulos de segurança com relação de dominância¸

clearance : S → L

classification : O → L

ATT(S) = {clearance}

ATT(O) = {classification}

allowed(s; o; read) ⇒ clearance(s) ≥ classification(o)

allowed(s; o;write) ⇒ clearance(s) ≤ classification(o)

Neste exemplo, os rótulos de segurança (*clearance* e *classification*) são usados como um atributo do sujeito e do objeto e as propriedades de segurança são utilizadas para pré-autorizações. Se a *clearance* do sujeito s domina a *classification* do objeto o, a solicitação de *read* é admitida. O processo é semelhante no *write*.

Exemplo 2 DAC utilizando políticas fechadas ACL com um ID individual, UCONpreA0 :

N é um conjunto de nomes identificadores

id : S → N, mapeamento de um para um

ACL : O → 2^{NxR}

ATT(S) = {id}

ATT(O) = {ACL}

allowed(s; o; r) => (id(s); r) ∈ ACL(o)

No exemplo do DAC, identidades individuais ou de grupo e a ACL são atributos do sujeito e do objeto respectivamente. A ACL é um mapeamento funcional do objeto para múltiplos ids e direitos. Se a identidade de um sujeito junto com o direito à requisição existe na ACL, a requisição é admitida.

No RBAC, a função do usuário e as permissões pertinentes a sua função podem ser considerados como atributos do sujeito e do objeto.

Modelos ongoing-obrigações - $UCON_{onB}$

São modelos similares ao UCONpreB, exceto pelas obrigações que tem serem realizadas enquanto os direitos são exercidos. Eles podem ser realizados de forma periódica ou de forma contínua. Para isto foi introduzido o parâmetro tempo T como partes das obrigações onOBL. O parâmetro T é definido como intervalos de tempo baseados em períodos ou eventos.

Por exemplo, um sujeito pode clicar em um aviso para ser executado em 20 dias (deseja se registrar agora ou mais tarde).

O modelo $UCON_{onB}$ possui os seguintes componentes:

O modelo UCONonB0 possui os seguintes componentes:

S;O;R;ATT(S);ATT(O);OBS;OBO; e OB não são mudados pelo UCONpreB;

T, é um conjunto de valores de tempo ou elementos deu m evento;

onB and onOBL, (os predicados ongoing-obligations e os elementos ongoing-obligation, respectivamente);

onOBL $\subseteq$ OBS x OBO x OB x T;

getOnOBL : S x O x R $\rightarrow 2^{onOBL}$, a função escolhe um ongoing-obrigations para uma resquisição de uso

onFulfilled : OBS x OBO x OB x T $\rightarrow$ {true; false};

onB(s; o; r) = $\bigwedge$(obs_i; obo_i; ob_i;t_i)$\in$getOnOBL(s;o;r) ^{onFulfilled}(obs_i; obo_i; ob_i;t_i);

onB(s; o; r) = true por definição se getOnOBL(s; o; r) = $\varnothing$;

allowed(s; o; r) $\Rightarrow$ true;

stopped(s; o; r) $\Leftarrow \neg$onB(s; o; r).
O modelo UCONonB1 é idêntico ao UCONonB0 exceto pela adição dos seguintes processos pré-atualizações:

preUpdate(ATT(s)); preUpdate(ATT(o)): um procedimento opcional para alterar alguns atributos como conseqüência das pré-obrigações.
O modelo UCONonB2 é idêntico ao UCONonB0 exceto pela adição dos seguintes processos ongoing-atualizações:

onUpdate(ATT(s)); onUpdate(ATT(o)): um procedimento opcional para alterar alguns atributos como conseqüência das pré-obrigações.
O modelo UCONonB3 é idêntico ao UCONonB0 exceto pela adição dos seguintes processos pós-atualizações:

postUpdate(ATT(s)); postUpdate(ATT(o um procedimento opcional para alterar alguns atributos como conseqüência das pré-obrigações.

Mostra-se um simples exemplo do UCONonB0 .

Visualiza-se um aviso no windpws enquanto s exerce r, UCONonB0 :

OBS = S

OBO = {ad window}

OB = {keep active}

T = {always}

getOnOBL(s; o; r) = {(s; ad window; keep active; always)}

allowed(s; o; r) $\Rightarrow$ true

stopped(s; o; r) $\Leftarrow \neg$onFulfilled(s; ad window; keep active; always)

Aqui, somente uma obrigação ongoing é requisitada. Suponha um provedor de serviço livre na Internet que solicita que os usuários vejam os avisos enquanto conectados ao servidor. Neste caso, não existe solicitação que tenha que ser completada antes do uso do serviço. Tão logo o aviso é ativado, o serviço é liberado.

Modelo pré-condições - $\mathbf{UCON_{preC}}$

Este modelo inclui algumas restrições ambientais que não estão diretamente relacionadas aos sujeitos e objetos. O ambiente atual e o status do sistema é retornado, e a cada vez que isto acontece a condição é avaliada. Via a utilização de condições no processo de decisão de uso, o UCONc pode prover controles finos de uso. Ao contrário dos modelos de autorização e obrigação, o de condição não pode ser mutável.

As seguintes definições formalizam o modelo $UCON_{preC:}$

O modelo UCONpreC0 possui os seguintes componentes:

S;O;R;ATT(S); and ATT(O) are not changed from UCONpreA;

preCON (a set of pre-conditions elements);

getP reCON : S x O x R $\rightarrow 2^{preCON}$;

preConChecked : preCON $\rightarrow$ {true; false};

preC(s; o; r) = $\wedge$ preCon$_i \in$ getP reCON(s;o;r) preConChecked(preCon$_i$)

allowed(s; o; r) $\Rightarrow$ preC(s; o; r).

No $UCON_{preC0}$, o preC é utilizado no processo de decisão de uso junto com S, O e R. O conjunto relevante de elementos de condição preCON é escolhido com base na possível requisição, usando os atributos do sujeito e do objeto. Para admitir uma requisição, todas as restrições de condição devem ser avaliadas.

Por exemplo, suponha que existam requisições para restringir localizações onde o uso pode ser exercido. Isto pode ser realizado, por exemplo, verificando o endereço IP antes do uso ser admitido ($UCON_{preC0).}$

Exemplo: limite de localização, UCONpreC0 :

studentAREA; facultyAREA (admite códigos de área para student e faculty) curArea is código de área de um dispositivo atual

ATT(s) = fmemberg

preCON = {(curArea $\in$ studentAREA); (curArea $\in$ facultyAREA)}

getP reCON(s; o; r) = (curArea 2 studentAREA); if member(s) =`student';

(curArea 2 facultyAREA); if member(s) =`faculty'.

allowed(s; o; r) $\Rightarrow$ preConChecked(getP reCON(s; o; r))

O exemplo verifica a localização corrente de um usuário em tempo de solicitação. As localizações admitidas pelo estudante e pela Universidade podem ser diferentes e tenha que se chegar a um acordo. Este exemplo assume que não existe mudança de localização enquanto a requisição é exercida ou não exista restrição de mudanças de localização durante o uso se a localização original tenha sido aprovada.

4.6 Modelos de pesquisa

A comunidade de segurança no âmbito mundial vem buscando melhorias nos procedimentos de controle de acesso. Esta seção apresenta alguns modelos desenvolvidos por pesquisadores que apresentam novas abordagens sobre o assunto.

4.6.1 Or-BAC (Organization Based Access Control)

O Or-BAC é um modelo de controle de acesso que tem o conceito de organização como a sua principal linha de ação. Este modelo não está restrito somente a garantir ou não permissões, mas possibilita estabelecer proibições, obrigações e recomendações.

O conceito de organização é visto como um grupo organizado de sujeitos representando um papel. Isto significa que o papel exercido por um sujeito corresponde a alguma relação de concordância entre os sujeitos na formação da organização.

4.6.1.1 O Modelo Or-BAC [Kalam, et. al. 2003]

Vários modelos são utilizados para compor o modelo Or-BAC como um todo. Todos possuem a sua representação sob a forma do modelo E-R (entidade – relacionamento), onde as suas entidades são relacionadas entre si dentro de uma organização. Por exemplo, o modelo sujeitos-papéis.

Sujeitos e papéis

Os sujeitos no modelo são entidades ativas, que podem ser representados por usuários (p. ex. Luiz, Ana, professor, aluno) ou organizações (p. ex. Departamento de ensino de uma Faculdade). Os papéis estruturam uma ligação entre os sujeitos e as organizações (professores horistas). A representação do relacionamento emprega no modelo Or-BAC é mostrado na Figura 4.16.

O modelo possibilita, via a entidade papel, estruturar os sujeitos e atualizar de forma simples a política de segurança quando novos sujeitos são adicionados ao sistema.

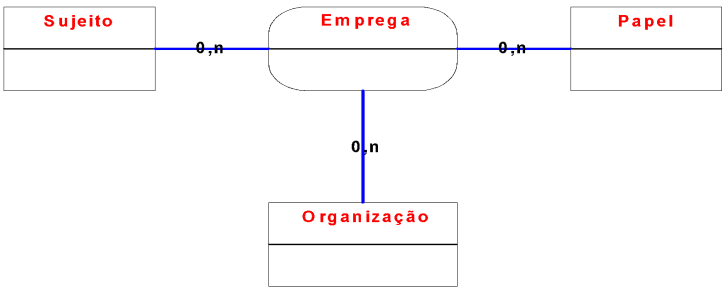

Figura 4.16. Modelo via entidade papel.

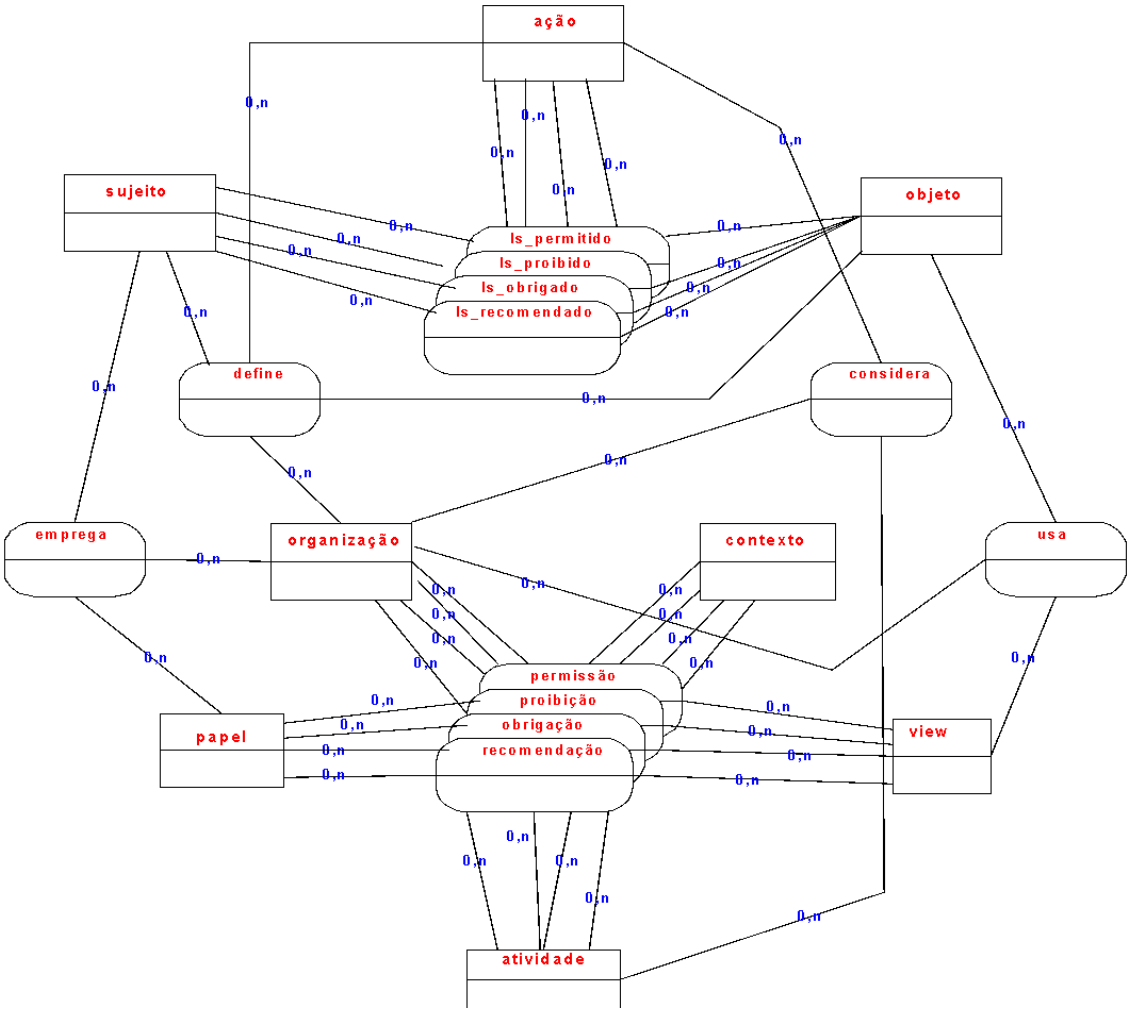

Figure 4.17. Modelo Or-BAC.

Autorização concreta

Apresentado todos os modelos que compõe o modelo Or-BAC (Figura 4.17), existe agora a possibilidade de modelar as permissões concretas. As permissões são chamadas desta forma porque foram introduzidos novos relacionamentos. Por exemplo, o conceito de Is_permitido como um relacionamento entre sujeitos (s), objetos (o) e ações (α) – Is_permitido(s, o, α), significando que um sujeito s pode realizar uma ação α em um objeto o. Os relacionamentos Is_proibido, Is_obrigado e Is_recomendado também são modelados de forma semelhante.

Cada instância do relacionamento Is_permitido é logicamente derivada das permissões garantidas para papéis, views e atividades obtidas do relacionamento Permissão. Sendo assim, este modelo faz da organização a sua abordagem, e o contexto a referência para a garantia do acesso do sujeito a um objeto na realização de uma ação.

4.6.2 TBAC – Tasked-based Authorization Controls

O TBAC é um modelo de controle de acesso que possui características diferentes dos controles de acesso tradicionais e modelos de segurança. Ele aborda a modelagem e a aplicação de segurança no nível dos serviços executados na empresa. Em função desta

nova abordagem, o TBAC cria um novo fundamento, o modelo de segurança ativo. Este fundamento trabalha a modelagem e aplicação de segurança sob a perspectiva de atividades ou serviços, provendo abstrações e mecanismos para ativar o gerenciamento de segurança, em tempo de execução, conforme a evolução dos serviços realizados. Outra característica do modelo de segurança ativo é a capacidade de gerenciamento ativo das permissões, elas são constantemente monitoradas, ativadas e desativadas conforme a evolução do contexto, isto é, em função dos serviços que estão sendo realizados [Thomas e Sandhu 1997, Thomas e Sandhu 1994].

Esta nova visão, segurança baseada na evolução de serviços, representa uma ruptura dos modelos de segurança clássicos, tal como aqueles baseados em uma ou mais variações da visão de segurança de controle de acesso entre sujeito e objeto e o controle centralizado das decisões de permissões. A flexibilidade de se obter autorizações baseadas em serviços (na sua evolução) possibilita uma maior agilidade nos processos de obtenção das permissões, não somente pela necessidade de se automatizar o processo de autorização e os controle de acesso relacionados, mas também pela ausência da figura do administrador de segurança na gerência durante a sua realização [Thomas e Sandhu 1997, Thomas e Sandhu 1993].

Para que se alcance realmente a agilidade e automatização proposta pelo modelo, a sua aplicabilidade deve ser direcionada a sistemas que provêem permissões curtas e em intervalos de tempo pré-determinados, principalmente em ambientes baseados em transações e *workflows*. Outro aspecto importante é o direcionamento para sistemas que tenham capacidade auto-administrativa, reduzindo a sobrecarga associada com a administração de segurança existente entre sujeito e objeto.

A utilização do TBAC em sistemas *workflows* possibilita que as permissões sejam garantidas, utilizadas e revogadas automaticamente, e coordenadas conforme a evolução dos diversos serviços realizados. Desta forma, se evita que as permissões sejam ativadas antes ou após surgir a necessidade dos serviços, ou que as mesmas estejam ainda ativas após os serviços terem sido encerrados, criando vulnerabilidades nos sistemas. O TBAC descarta também a responsabilidade do administrador de segurança manter um controle constante da evolução dos serviços e com isso as permissões compatíveis para as suas execuções. É claro que as permissões de controle de acesso, apesar de automatizadas, seguem a lógica da aplicação, bem como uma política de controle de acesso pré-estabelecida.

4.6.2.1 TBAC como modelo de segurança ativo

O conceito de modelo de segurança ativo caracteriza os modelos que reconhecem todo o contexto em que as requisições de segurança aparecem, e aplica a atividade de gerenciamento de segurança conforme a evolução do contexto, com base no progresso das atividades realizadas.

A Figura 4.18 apresenta o progresso da autorização (*Authorization-step*), considerada a mais fundamental abstração do TBAC, isto porque a autorização é fornecida passo a passo conforme a evolução da execução das tarefas (a autorização é fornecida conforme a necessidade de uso). Ela representa o progresso do processamento inicial da autorização, agrupando sujeitos (*executor*) confiáveis a um conjunto de permissões, semelhante no mundo do papel (ex: RBAC), onde um usuário ou um grupo de usuários

pode ser agrupado conforme o papel exercido e estar ligado a um conjunto de funções que podem ser exercidas (ex: o gerente de vendas Marco pode autorizar a ordem de venda X) [Thomas e Sandhu 1997].

O progresso da autorização no TBAC está associado com o grupo de *trustee* (depositário) chamado de *trustee-set*. Um membro deste conjunto (*executor-trustee* daquele passo) irá eventualmente permitir o progresso da autorização quando ele for instanciado. As permissões necessárias ao *executor-trustee* para invocar e permitir o progresso de autorização compõe um conjunto de permissões chamada de permissões do *executor*. As permissões que são habilitadas no progresso de autorização formam o conjunto de permissões habilitadas, e a união das permissões do *executor* e das permissões habilitadas é conhecida como estado de proteção do progresso de autorização. O período de validade e o ciclo de vida da atividade estão associados com toda o progresso de autorização.

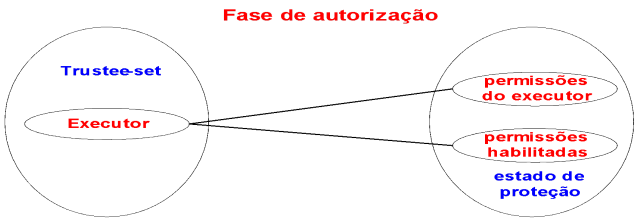

Figura 4.18. Authorization-step.

O TBAC difere dos modelos de controle de acesso tradicionais pela inclusão do domínio de progressos de autorização e do domínio de estimativa de validade e uso, que embutem a informação contextual baseada em tarefas.

A Figura 4.19 [Thomas e Sandhu 1997] apresenta os conceitos, características e componentes que fazem o TBAC um modelo de segurança ativo, como:

> Modelagem de autorização em serviços e workflows, bem como o monitoramento e gerenciamento do processamento da autorização e ciclos de vida como progresso das tarefas;
>
> O uso do controle de acesso baseado no uso e no tipo;
>
> A manutenção de proteções separadas para cada progresso de autorização; e
>
> Execução dinâmica dos procedimentos de entrada e saída das permissões dos estados de proteção como progressos de autorização são processadas.

Todo progresso da autorização mantém o seu próprio estado de proteção. O valor inicial do estado de proteção é o conjunto de permissões que são ativadas como resultado da validação do progresso da autorização (*authorization-step*). Portanto, o conteúdo deste conjunto de permissões sofrerá mudanças conforme o progresso da autorização é processado e as permissões forem sendo utilizadas. Para cada permissão atrela-se uma estimativa de uso. Quando se alcança esta estimativa, a permissão associada é desativada e a atividade correspondente não é mais permitida. A constante verificação automática dos procedimentos de entrada e saída das permissões como autorizações é a principal característica que torna o modelo TBAC ativo. Além desta questão, existe o

aspecto de que toda a permissão no estado de proteção é única, e mapeada para uma instância do progresso da autorização e para uma tarefa que está invocando a autorização.

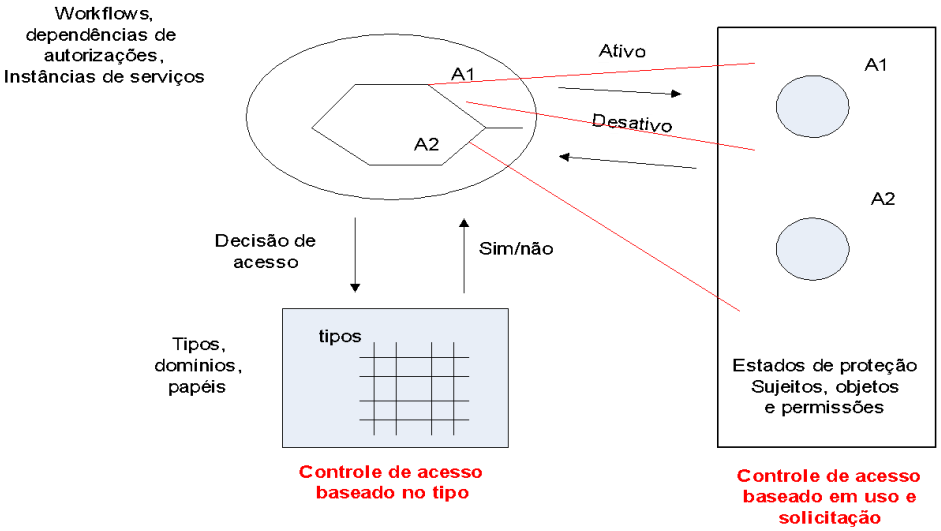

Figura 4.19. TBAC como modelo de segurança ativo.

As diferenças entre os controles de acesso baseado em uso e solicitação, e baseado em tipo é também uma característica significante do modelo TBAC. O controle de acesso baseado em tipo é utilizado para encapsular restrições de controle de acesso especificadas pela política de controle de acesso e aplicadas a tipos. Controle de acesso baseado em uso e solicitação é usado para modelar e gerenciar os detalhes do controle de acesso e as permissões das solicitações de autorização individuais, incluindo a manutenção do uso de permissões.

4.6.2.2 A família dos modelos TBAC

A família dos modelos TBAC possui como base o modelo $TBAC_0$, sendo um modelo que oferece algumas facilidades para modelar serviços, progressos da autorização e dependências, relatando vários progressos da autorização. É considerado o modelo mais genérico e flexível. Os modelos avançados $TBAC_1$ e $TBAC_2$ herdam as características de $TBAC_0$, mas também incorporam as suas próprias características. O $TBAC_1$ incorpora a noção de autorizações compostas e o $TBAC_2$ a questão de restrições. O $TBAC_3$ incorpora as funcionalidades dos três modelos anteriores.

4.6.3 TMAC – Team-based Access Control [Thomas 1997]

O TMAC é uma abordagem para aplicação do controle de acesso baseado em papel em ambientes colaborativos tais como aqueles que envolvem *workflow*. A abordagem em atividades colaborativas deve-se ao fato de se obter a perfeição via o uso de grupos organizados. Logo, o aspecto central da abordagem TMAC é a noção de grupo como uma abstração que encapsula uma grupo de usuários com o objetivo de executar um determinado serviço ou alcançar uma determinada meta.

O trabalho desenvolvido por esta abordagem de controle de acesso visa criar um paradigma de segurança, apresentando melhorias de segurança para um *workflow* de uma Empresa. Três objetivos são buscados nesta abordagem:

- Criar um ambiente seguro que não possa ser invadido pelo quadro de funcionários;
- Prover um sistema rígido quanto ao acesso a determinada informação de um segmento de negócio em um período de tempo; e
- Criar uma infra-estrutura de segurança que não necessite de um grande volume administrativo, isto é, possa ser difícil de administrar na sua maioria.

Estas questões podem ser mais bem entendidas via a apresentação dos modelos de segurança passivo e ativo. O modelo de segurança ativo é aquele que tem inicialmente a função da manutenção das atribuições de permissão, tal como no RBAC, onde as permissões são atribuídas a papéis, e distingue as ativações de permissão baseada em contexto e tarefas. Sendo assim, após a permissão ser atribuída, ela pode ser ativada ou desativada várias vezes de acordo com o contexto o qual está associado com a evolução de execução das tarefas.

No modelo passivo, a permissão é atribuída somente ao usuário. Assume-se sempre que ela pode ser ativada independente de qualquer outra consideração como o contexto. Este é o típico caso da obtenção da permissão com base na lista de controle de acesso (ACL).

4.6.3.1 Apresentação do Modelo

O TMAC trabalha com controle de acesso baseado em papel em ambientes colaborativos (*workflow*). Dois aspectos são necessários para o controle de acesso em atividades colaborativas:

1. a necessidade da permissão baseada em papel; e
2. a necessidade de se ter a ativação da permissão no nível de usuário e objetos individualmente.

O TMAC cria a abordagem de controle de acesso com as duas atividades, acima citada, trabalhando de forma concorrente. Para que isto aconteça necessita-se de:

> Uma abstração para limitar e modelar um conjunto de usuários e os seus respectivos papéis;
>
> Registro de memória de todo o contexto de colaboração para um conjunto de usuários.

Nos modelos RBAC, um grupo de usuários está relacionado a um papel, limitando o trabalho em equipe. O TMAC, por visar o conceito de euipe, introduz o conceito de contexto de colaboração. Este contexto contém a informação sobre toda as tarefas que serão executadas. Do ponto de vista de controle de acesso, o contexto de colaboração da equipe pode conter dois tipos de informação: contexto do usuário - os usuários que compõe a equipe; e contexto do objeto – o conjunto de instância de objetos necessária pela equipe para realizar a sua tarefa. Portanto, conhecendo a estrutura básica da equipe

em termos de seus vários papéis, encontra-se o aspecto 1, citado anteriormente, e conhecendo o contexto de colaboração encontra-se o aspecto 2.

A Figura 4.20 apresenta os principais conceitos do TMAC, e a interação entre eles.

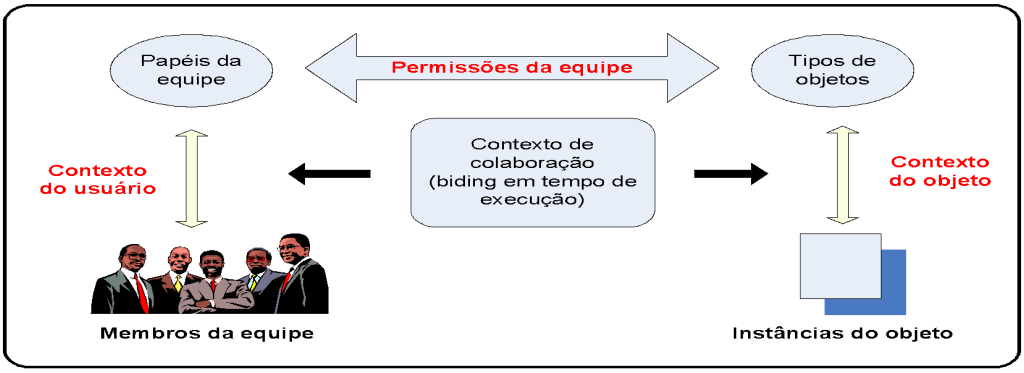

Figura 4.20. Principais conceitos do TMAC.

Uma equipe na abordagem TMAC consiste de:

O nome da equipe, T;

O conjunto de membros/usuários da equipe, TU;

O conjunto de papéis da equipe, TR – TR $\subseteq$ R, onde R é o conjunto total de papéis no sistema de informação;

O papel chefe da equipe (h), onde h $\subset$ TR. Somente um usuário pode ser o chefe da equipe a cada vez;

O conjunto dos tipos de objetos, OT;

O conjunto de instâncias de objetos, O;

O conjunto de permissões da equipe, TP, definida via TR e OT, isto é, TP $\subseteq$ TR x OT;

O contexto de colaboração consiste de dois componentes:

O contexto do usuário (UC), onde UC:TR x TU;

O contexto do objeto (OC), onde OC:OT x O.

A idéia básica no TMAC é usar o RBAC para definir o conjunto de permissões P via os domínios de R e OT. Equipes individuais de mesma estrutura (tipo/classe) englobará o mesmo subconjunto de papéis, TR de R e assim herdará o mesmo subconjunto TP e P. Porém, o TMAC invoca em tempo de execução a ligação do TP de cada equipe para os conjuntos TU e O da equipe. Isto admite a ativação em tempo de execução das permissões no nível de usuários e objetos individuais.

Sob o ponto de vista operacional e de implementação, o TMAC poderia suportar as seguintes primitivas para habilitar o controle de acesso na equipe como um todo:

- Usuário_atribuição (usuário, equipe): atribuição de um usuário a uma equipe;
- Usuário_desabilita (usuário, equipe): o usuário deixa de ser da equipe;

- Equipe_ativa (equipe): amarra as permissões da equipe aos membros da equipe e os objetos necessários (TU e O); e
- Equipe_desativa (equipe): desativas as permissões para toda a equipe.

Nas aplicações a ativação e desativação das permissões são realizadas por papel, mas podem existir casos onde elas são ativadas e desativadas usuário por usuário. Neste caso as primitivas de ativa_usuário e desativa_usuário podem ser habilitadas.

Para finalizar o TMAC pode ser auto-administrado via chamadas básicas emitidas pelo sistema de informação, atribuindo ou não os membros de uma equipe, em tempo de execução conforme a evolução do *workflow*. Esta ativação pode ser sincronizada com as primitivas de atribuição e ativação do usuário para automatizar a administração de segurança. Para preservar o controle de acesso a objetos individual via equipes, o contexto do objeto pode ser passado de uma equipe para outra.

Sendo assim, o TMAC possibilita formular um modelo de segurança que trata a natureza dos acessos baseado em equipe e trabalha de forma colaborativa. Ele possui a vantagem de ser capaz melhor administrar e modelar o sistema, via um maior refinamento no controle sobre ativação das permissões para objetos e usuários individuais.

4.7 Aspectos comparativos

Como visto durante o mini-curso, todos os modelos de controle de acesso visam minimizar os problemas de segurança estabelecendo o que, quando e como acessar, com base em uma política de controle de acesso, mas cada um com a sua abordagem. Esta seção busca apresentar uma pequena comparação entre estas diversas formas de abordagens, focando os seus aspectos positivos e ou negativos.

Os modelos de controle de acesso tradicionais como RBAC, MAC e DAC consideram inicialmente as decisões de autorizações estáticas baseadas nas permissões de sujeitos em objetos. Eles também possuem sistemas de gerenciamento de autorização baseados em políticas, possuidores de um monitor de referência centralizado ou distribuído, com administração centralizada, que verifica as permissões de cada sujeito quando este requisita o acesso. A permissão é garantida ao sujeito conforme a política de segurança em tempo de requisição de acesso, e não existe um limite de tempo de uso do objeto.

O DAC é um controle de acesso simples de ser implementado, pois faz uso de ACLs, e é implementado na maioria dos sistemas operacionais existentes, tornando a sua utilização bastante divulgada na comunidade de segurança. Porém, este modelo possui falhas quanto aos aspectos de segurança. Estas falhas são causadas devido a possibilidade da passagem de permissão sem um conhecimento prévio de seu dono ou do gerente de segurança, provendo vulnerabilidades de acesso aos objetos do sistema. Arquivos que possuam informações sensíveis podem tramitar entre sujeitos que não estejam qualificados a acessá-los. Portanto, o DAC não é um controle de acesso que possui características que garantam de forma efetiva o acesso somente a sujeitos que realmente possuam direito de acesso.

O MAC pode estender ou substituir o DAC por permissões de sistemas de arquivos os conceitos de usuários e grupos. É uma estrutura mais complexa de ser implementada,

vista em alguns sistemas (Free BSD), porém mais segura que o DAC. Esta maior segurança deve-se ao aspecto do sujeito não possuir mais o controle completo do acesso aos recursos por ele criado, e a política de segurança do sistema (especificado pelo administrador de segurança) determina inteiramente o acesso que deve ser concedido aos objetos pelos sujeitos. A possibilidade de classificar o sujeito e o objeto quanto ao grau de sensibilidade da informação foi um grande avanço para o controle de acesso. Esta classificação viabilizou o princípio da distribuição dos sujeitos e objetos em categorias, isto é, relacionou o acesso de sujeitos a um conjunto de categorias de objetos conforme o seu grau de confidencialidade. Sendo assim, o MAC pode ser considerado um controle de acesso rígido, sendo aplicado em organizações onde informações são classificadas como sensíveis. A sua grande desvantagem de da capacidade de classificação da política de segurança, que está atrelada diretamente a capacidade de seu criador.

O RBAC, apesar de ser uma boa opção de controle de acesso, ele não atende as questões de ambientes colaborativos como:

> a necessidade de um controle de acesso híbrido que incorpore as vantagens de se ter um amplo leque de permissões baseado em papéis via tipos de objetos e um controle refinado de determinados usuários em determinados papéis e em instâncias individuais de objetos;
>
> a necessidade de reconhecer o contexto associado com as tarefas colaborativas e a habilidade para aplicar este contexto em decisões na ativação de permissões.

O RBAC é um modelo de extrema aplicabilidade em organizações, pois o uso do conceito de papel facilita e simplifica a criação e gerenciamento da política de segurança. Junto com o conceito de papel a propriedade de hierarquia também colabora com estes aspectos. Porém, a facilidade do uso do papel e da hierarquia pode trazer problemas quanto a segurança. A herança obtida por papéis subordinados pode causar problemas de segurança, isto é, ter acesso a objetos não adequados ao sujeito. O administrador de segurança deve estar atento, em conjunto com a política de segurança especificada, quanto às questões de limitação de acesso pelos sujeitos. Mesmo sendo a característica de herança um fator positivo, ela deve ser cuidadosamente analisada de forma que acessos indesejados a objetos sejam a sujeitos.

Controles de acesso como o DRM também seguem a mesma linha de ação dos controles DAC, MAC e RBAC. Possuem os seus monitores de referências distribuídos conforme a distribuição dos produtos pelos fornecedores, mas possuem problemas de segurança. O modelo de controle de acesso DRM busca cada vez mais limitar o acesso não autorizado a objetos remotos. Porém, o DRM apresenta normalmente um grande problema: os usuários dos objetos podem fazer uso de forma indiscriminada tanto a sua distribuição quanto ao seu tempo de uso. O DRM procura garantir a segurança nos servidores provedores de produtos, mas na sua maioria não tem condições de garantir que os usuários possuam ambientes seguros ou que estes façam bom uso dos produtos. Sendo assim, estes fatores são aspectos limitadores no DRM, que devem ser trabalhados pelos seus desenvolvedores.

Com o desenvolvimento da tecnologia da informação, principalmente no comércio eletrônico, alguns aspectos adicionais tornaram-se necessários ao controle de acesso. Os sistemas de informação, que usam objetos digitais, passaram a se preocupar com os aspectos temporais de utilização de um objeto (ex: o tempo de utilização de um objeto é consumido conforme a sua utilização). Logo, a permissão de um sujeito a um objeto deve ser atualizada em tempo de execução conforme a sua utilização, até expirar totalmente, além poder revogá-la durante o uso do objeto. O UCON é um modelo que veio revolucionar e resolver diversos problemas já citados. Tem como principais vantagens o controle dinâmico do uso do objeto (nenhum outro modelo possui) via a capacidade de mutabilidade de atributos. Esta característica acaba com o problema do uso eterno do objeto, passando a estabelecer limites de tempo na sua utilização. Vale também ressaltar a capacidade de operar com os modelos DAC, MAC, RBAC e DRM, por exemplo, possibilitando uma maior versatilidade das suas funcionalidades.

Os demais modelos citados no mini-curso buscam novas abordagens não mais restritas em garantir ou não a permissão, ou mesmo preocupados com a relação sujeito, objeto e permissão. O Or-BAC baseia-se no conceito de organização, e trabalha com os aspectos de proibições, obrigações e recomendações. Usa o conceito de contexto como a sua grande arma. A sua abordagem estabelece que o papel exercido por um sujeito corresponde a alguma relação de concordância entre os sujeitos na formação da organização. Sendo assim, todos os aspectos de estabelecimento de permissões ou proibições está baseado no contexto em que o sujeito ou grupo se encontram na organização e na execução das suas tarefas. O Or-BAC consegue estabelecer um controle mais amplo dos papéis dentro da organização, mas deixa de existir um controle efetivo de cada sujeito em relação a um objeto, ocasionando possíveis problemas de gerenciamento da política de controle de acesso pelo administrador de segurança.

Os controles de acesso TMAC e TBAC, como citado no parágrafo anterior, também fogem ao escopo da permissão direta entre sujeito e objeto. Eles trazem uma grande vantagem em relação aos demais e ambos são auto-administráveis, possibilitando um maior dinamismo e rapidez nas tomadas de decisão. O gerente de segurança programa a política de controle de acesso uma única vez, e esta é aplicada conforme com a evolução das tarefas dentro do sistema.

Esta vantagem pode talvez proporcionar problemas não quanto ao desempenho, mas sim quanto ao gerenciamento de um incidente de segurança. Este problema deve-se a necessidade de se criar pontos de controle ou sistemáticas que verifiquem ou que garantam que a política de controle de acesso aplicada é adequada as necessidades da organização.

Após esta breve análise dos modelos de controle de acesso apresentados neste minicurso, pode-se observar que o controle de acesso sofreu uma evolução no decorrer do tempo, principalmente em função das novas necessidades apresentadas também pela evolução dos sistemas computacionais. Todos eles podem ser utilizados, sozinhos ou em conjunto, conforme a necessidade de segurança das organizações, mas seja cuidadoso quanto a determinação da sua política de segurança e quanto a escolha do modelo, porque mal utilizados podem reverter a sua função de segurança a uma ferramenta de ajuda a uma invasão ao seu sistema computacional.

4.8 Conclusão

O controle de acesso é um serviço de segurança e tem como função gerenciar o acesso aos objetos de seu sistema computacional. Este mini-curso apresentou as características fundamentais do controle de acesso e alguns dos seus principais modelos. Iniciou-se com os modelos mais tradicionais como DAC e MAC, apresentando as suas principais características e funcionalidades. O RBAC inclui o conceito de papel, que possibilitou uma maior simplicidade e facilidade quanto a modelagem e a administração do controle de acesso dentro da organização. A possibilidade de estabelecer o acesso aos objetos com base no papel foi uma revolução frente ao DAC e MAC. O DRM mostrou como podem ser realizados os acessos aos recursos digitais providos por um fornecedor garantindo os direitos autorais do sistema.

Fugindo aos modelos tradicionais de controle de acesso o UCON veio como o grande modelo inovador, trazendo um maior dinamismo em relação aos controles de acesso anteriores (mutabilidade de atributos, aspectos temporais). A possibilidade de incorporar as funcionalidades dos modelos como DAC, MAC, DRM e RBAC deu ao UCON um escopo maior de abrangência e hoje pode ser considerado o modelo mais completo.

Foram também apresentados os modelos desenvolvidos pela comunidade de pesquisa na área de segurança. Alguns destes modelos trabalham de forma que o administrador do sistema não atue diretamente no processo administrativo do controle de acesso. Os modelos são auto-administrados, no qual o administrador de segurança atua somente no início do processo (TMAC e TBAC). O Or-BAC traz o contexto como a sua grande virtude, e com isso possibilita limitar o acesso a determinadas situações dentro das organizações. Com base no que foi apresentado pode-se dizer que cada um dos modelos apresentados possui os seus valores e as suas aplicabilidades dentro de um sistema computacional. Não se pode dizer que eles irão resolver todos os problemas de controle de acesso aos recursos de um sistema, mas com certeza eles reduzirão sensivelmente.

4.10 Referências

Amoroso, Edward G. (1994) “Fundamentals of Computer Security Technology”, Prentice Hall PTR, Upper Saddle River, NJ.

Anderson, James P. (1972) ”Computer Security Technology Planning Study” Report ESD-TR-73-51.Electronic Systems Division.

Bechtold S. (2001) “Implications of Digital rights management, security and privacy in Digital rights management”, Proceedings of ACM - Workshop DRM p. 213 – 232.

Bell, D. E, e LaPadula, Leonard J. (1976) “Secure Computer Systems: Unified Exposition and Multics Interpretation”, MITRE Technical Report MTR-2997 Rev. 1, MITRE Corporation.

Biba, Kenneth J. (1977) “Integrity Considerations for Secure Computer Systems”, MITRE Technical Report MTR-3153, MITRE Corporation, Bedford, MA.

Bishop, M. (2003) “Computer Security Art and Sciense”, ed. Addison Wesley

Buenett, S. e Paine, S. (2002) “Criptografia e segurança”, Ed. Campus.

Camelot (2001) “Differentiating Between Access Control Terms” Network Security Library :: Auth. & Access Control.

Clark, David D. e Wilson, David R. (1987) "A Comparison of Commercial and Military Computer Security Policies", In Proceedings of the IEEE Symposium on Security and Privacy, p. 184– 194, Oakland, CA.

Cuppens, F. e Miège, A. (2003) "Administration Model for Or-BAC", Workshop on Metadata for Security (WMS).

Cuppens, F. e Miège, A. (2003) "Modelling Contexts in the Or-BAC Model", Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003), IEEE Press.

Curphey, M., Endler, D., Hau, W. e Taylor S. (2002) "A Guide to Building Secure Web Applications - Mandatory Access Control – Chapter 8". Access Control and Authorization, The Open Web Application Security Project (OWASP).

Denning, D. E. R. (1982) "Cryptography and data security", Addison-Wesley.

Departament of Defense (1985). "Trusted Computer System Evaluation Criteria", DOD 5200.28-STD.

Duncan C., Barker E., Peter D., Morrey M. e Waelde C. (2004) "Digital Rights Management", JISC DRM Study – Final Report.

El Kalam, A. A., El Baida, R., Balbiani P., Benferhat S., Cuppens F., Deswarte Y., Miège A., Saurel C. e Trouessin, G. (2003) "Organization based access control", Proceedings oh the 4th International on Policies for Distributed Systems and Networks, IEEE Press.

Ferraiolo, David F., Sandhu, Ravi S., Gavrila, S., Kuhn, D. R. e Chandramouli, R. (2001) "Proposed NIST Standard for Role-Based Access Control", ACM Transactions on Information and System Security, Vol. 4, No. 3, p. 224–274.

Goguen J. A. e Mesajuer J. (1982) "Security Policies And Security Models", Proceedings of IEEE symposium on Reseach in Security and Privacy.

Harrison, Michael A. e Ruzzo, Walter L. (1976) "Protection in Operating Systems", Communications of the ACM, Vol. 19, No 8.

Harrison, Michael A. Harrison, Ruzzo, Walter L. e Ullman, Jeffrey D. (1976) "Protection in Operating Systems", Communications of the ACM.

ISO/IEC 27001 (2005)"Tecnologia da Informação – Técnicas de Segurança – Sistemas de Gerenciamento de Segurança da Informação – Necessidades", ISO/IEC.

Jagadeesan, R. e Saraswat, V. (2005) "Timed Constraint Programing: A decarative Approach to Usage Control", Principles and Practice of Declarative Programming (PPDP'05).

Jordan, Carole S., Downs D., Wagner G., LaFountain, S. e Baker, Dixie B. (1987) "A Guide to Understanding Discrentionary Access Control in Trusted Systems", National Computer Security Center.

Kaminsky, Omar (2004) "Introdução à Gestão de Direitos Digitais", www.cem.itesm.mx/verba-iuris/articulos/080203.htm.

Karp, A. H. (2006) "Authorization-Based Access Control for the Services Oriented Architecture", 4th ICCC, IEEE Press

Katzenbeisser, Adelsbach, S. e Veith, H. (2003) "Watermarking schemes provably secure against copy and ambiguity attacks", Proceedings of the 2003 ACM workshop on Digital rights management, p. 111-119.

Ku, W. e Chi, Chi-Hung (2004) "Survey on the technological aspects of Digital Rights Management", Proceeding of the 7th Information Security Conference.

Lamport, L. (1994) "Transactions on Programming Languages and Systems - The Temporal Logic of Actions", ACM, Vol. 16 Issue 3.

Lampson, Butler W. (1971) "Protection"; Proceedings of the 5th Princeton Conference on Information Sciences and Systems, Princeton, p.437.

Landwehr, Carl E. (1981) "Formal Models for Computer Security", ACM Computing Surveys, 13(3): p. 247–278.

Landwehr Carl E, (1983) "Best available technologies for computer security", IEEE Comput, p.86-100

Landwehr, Carl E. (2001) "Computer security" Publicado por Springer-Verlag.

Mackenzie, D. e Pottinger, G. (1997) "Mathematics, Technology, and Trust: Formal Verification, Computer Security, and the U.S. Military", IEEE Annals of the History of Computing, Vol. 19, nr 3.

McLean, John (1990). "The Specification and Modeling of Computer Security". IEEE Computer, 23(1): p. 9–16.

Nicomette, Vincent (1996). "La Protection dans les Systèmes à Objets Répartis". Thèse de doctorat, Institut National Polytechnique de Toulouse, France.

Osborn, S. (1997) "Mandatory Access Control and Role-Based Access Control Revisited", Proceedings RBAC97.

Russel, Deborah e Gangemi, G. T. (1991) "Computer Security Basics", Ed. O' Reilly.

Samarati, Pierangela e Capitani di Vimercati, S. (2001)"Access Control: Policies, Models, and Mechanisms", Eds. R. Focardi and R. Gorrieri : FOSAD 2000, LNCS 2171, pp. 137–196.

Sandhu, Ravi S – Role (1997) "Based Access Control", SBC97.

Sandhu, Ravi S. (1993) "Lattice-Based Access Control Models", IEEE Computer, 26(11):p.9–19.

Sandhu, Ravi S. e Park, J. e Zhang X (2004) "Attribute Mutability in Usage Control", www.list.gmu.edu/confrnc/ifip/IFIP04-mutability.pdf.

Sandhu, Ravi S. e Park, Jaehong (2004) "The $UCON_{ABC}$ Usage Control Model", ACM Transactions on Information and System Security, Vol. 0, No. 0.

Sandhu, Ravi S. e Samarati, P. (1994) "Access Control: Principles and Practice", IEEE Communications Magazine.

Sandhu, Ravi S. e Samarati, P. (1996) "Authentication, Access Control, and Audit", ACM Computing Surveys, Vol. 28, No. 1.

Snyder, L. (1981) "Theft and Conspiracy in the Take-Grant Protection Model", Journal of Computerand System Sciences, p. 333–347.

The International DOI Foundation (2004) "DOI. The Digital Object Identifier system", http://www.doi.org/about_the_doi.html

The SCO Group (2004) "UnixWare 7 Documentation – Managing system security", http://ou800doc.caldera.com/en/SEC_admin/_Access_Control.html.

Thomas, R. K. e Sandhu, Ravi S (1993) “Towards a task-based paradigm for flexible and adaptable access control in distributed applications”, Proceedings of the Second New Security Paradigms Workshop, Little Compton, Rhode Island, IEEE Press.

Thomas, R. K. e Sandhu, Ravi S. (1997) “Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management”, Proceedings of the IFIP, Workshop on Database Security.

Thomas, R.K. and Sandhu, R.S. (1994) “Conceptual Foundations for A Model of Taskbased Authorizations”, Proceedings of the IEEE Computer Security Foundations Workshop, New Hampshire, IEEE Press.

Thomas, Roshan K. (1997) “Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments”, RBAC97

Thomas, T. (1988) “A Mandatory Access Control Mechanism for the Unix File System”, IEEE Press.

Yi C., Zhi-rong, Z. e Chang-xiang, S.(2002) “Design and Implementation MAC in Security Operating System”, Proceedings of IEEE TECON 02.

