



Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

CARLA MERKLE WESTPHALL

Florianópolis, Maio de 2005

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

---

Raul Sidnei Wazlawick

Banca Examinadora

---

Carla Merkle Westphall

---

Elias Procópio Duarte Júnior

---

Alexandre Moraes Ramos

---

Vitório Bruno Mazzola

Grande parte desta minha conquista tem como responsável minha esposa Thays, que com muito amor sempre esteve ao meu lado nos momentos difíceis de estudo e trabalho, me ajudando a alcançar meus objetivos.

Sem a confiança e o auxílio de meus pais Arisoly e Marlene eu também não estaria onde estou hoje. Minha formação moral e intelectual eu devo a eles, que sempre acreditaram e investiram em mim.

Grande parte do sucesso de um orientando está na qualidade de seu orientador. Sou muito grato à minha orientadora Carla por ter norteado meu trabalho com muita dedicação e paciência. Seus ensinamentos foram de muita importância para minha formação profissional.

Meus familiares, amigos e colegas de mestrado também tiveram muita importância nestes dois anos de mestrado. A todos eles vai o meu muito obrigado.

1.1.	Justificativa.....	19
1.2.	Objetivo Geral .....	20
1.3.	Objetivos Específicos .....	20
1.4.	Organização da Dissertação.....	20
2.1.	Introdução.....	22
2.1.1.	Componentes Adicionais de Segurança .....	23
2.1.2.	Políticas, Modelos e Mecanismos de Controle de Acesso .....	24
2.2.	Modelos Tradicionais de Controle de Acesso .....	25
2.2.1.	Modelo Discricionário.....	25
2.2.1.1.	Matriz de Acesso .....	26
2.2.2.	Modelo Obrigatório .....	29
2.2.3.	Modelo Baseado em Papel .....	31
2.2.3.1.	Descrição do Modelo RBAC.....	32
2.2.3.2.	Padronização do RBAC.....	33

2.2.3.3.	Previsões para o Futuro do Modelo RBAC .....	34
2.3.	Conceitos Modernos de Controle de Acesso .....	34
2.3.1.	Gerenciamento de Confiança.....	35
2.3.2.	Gerenciamento de Direitos Digitais .....	36
2.3.2.1.	Forma de Funcionamento .....	37
2.4.	Conclusões do Capítulo .....	40
3.1.	Introdução.....	41
3.1.1.	Novas Características do UCON .....	41
3.2.	Modelo UCON <sub>ABC</sub> .....	44
3.2.1.	Componentes do Modelo UCON <sub>ABC</sub> .....	44
3.2.1.1.	Sujeitos (S) .....	45
3.2.1.2.	Atributos do Sujeito ( ATT(S) ) .....	45
3.2.1.3.	Objetos (O) .....	46
3.2.1.4.	Atributos do Objeto ( ATT(O) ) .....	46
3.2.1.5.	Direitos (D).....	46
3.2.1.6.	Autorizações (A).....	47
3.2.1.7.	Obrigações (B).....	47
3.2.1.8.	Condições (C).....	48
3.2.2.	Modelos da Família UCON <sub>ABC</sub> .....	48
3.2.2.1.	Modelos de pré-Autorização - UCON <sub>preA</sub> .....	50
3.2.2.2.	Modelos de Autorização Durante a Execução - UCON <sub>onA</sub> .....	50
3.2.2.3.	Modelos de pré-Obrigaçãõ - UCON <sub>preB</sub> .....	51
3.2.2.4.	Modelos de Obrigaçãõ Durante a Execuçãõ - UCON <sub>onB</sub> .....	52
3.2.2.5.	Modelo de pré-Condiçãõ - UCON <sub>preC</sub> .....	53
3.2.2.6.	Modelo de Condiçãõ Durante a Execuçãõ – UCON <sub>onC</sub> .....	53
3.3.	Arquiteturas UCON.....	54
3.3.1.	Arquiteturas Relacionadas a Pagamento .....	54

3.3.1.1.	Tipo Baseado em Pagamento - PBT.....	55
3.3.1.2.	Tipo Livre de Pagamento - PFT .....	55
3.3.2.	Arquiteturas Baseadas na Localização do Monitor de Referência.....	55
3.3.2.1.	Monitor de Referência do Lado do Servidor - SRM .....	57
3.3.2.2.	Monitor de Referência do Lado do Cliente - CRM .....	57
3.3.2.3.	Monitor de Referência dos Lados Servidor e Cliente – SRM & CRM .....	58
3.4.	Conclusões do Capítulo .....	58
4.2.	<i>E-Marketplaces</i> .....	62
4.3.	Interação em Sistemas B2B.....	63
4.3.1.	Fatores de Avaliação de Sistemas B2B .....	63
4.3.2.	Camadas de Interação de Sistemas B2B.....	65
4.3.2.1.	Camada de Comunicação .....	65
4.3.2.2.	Camada de Conteúdo.....	66
4.3.2.3.	Camada de Processos Empresariais.....	67
4.4.	Conclusão do Capítulo .....	69
5.1.	Controle de Acesso Baseado em Política .....	70
5.1.1.	Relacionamentos.....	71
5.1.2.	Propriedade.....	72
5.1.3.	Políticas .....	72
5.1.4.	Gerenciador de Política .....	72
5.1.5.	Funcionamento do Sistema.....	73
5.2.	Permissão de Controle de Acesso Baseado em Lista .....	74
5.3.	Generic Authorization Mechanisms for Multi-Tier Applications - GAMMA.....	75
5.4.	Controle de Acesso Distribuído para Web Services.....	80

5.5.	Conclusão do Capítulo .....	81
6.1.	Modelo UCON <sub>ABC</sub> em Sistemas B2B .....	83
6.2.	Proposta .....	84
6.2.1.	Composição da Proposta .....	84
6.2.1.1.	Sistema da Empresa Provedora (SEP).....	84
6.2.1.2.	Sistema da Empresa Consumidora (SEC) .....	85
6.2.2.	Funcionamento .....	86
6.2.2.1.	Autenticação .....	86
6.2.2.2.	Controle de Acesso.....	86
6.2.3.	Gerenciamento de Permissões.....	88
6.2.3.1.	Descrição do Agrupamento Implícito - AI.....	88
6.2.3.2.	Desvantagem do Agrupamento Implícito.....	91
6.2.3.3.	Agrupamento Implícito Parcial - AIP.....	93
6.2.3.4.	Inclusão do Agrupamento Implícito Parcial na Proposta de Aplicação.....	94
6.2.4.	Arquiteturas Utilizadas Pela Proposta .....	94
6.3.	Sujeito Composto .....	95
6.4.	Comparação da Proposta com Trabalhos Relacionados.....	96
6.5.	Conclusão do Capítulo .....	97
7.1.	Tecnologias Utilizadas .....	98
7.1.1.	Linguagem Java.....	98
7.1.1.1.	Plataforma Java 2 Enterprise Edition (J2EE™).....	98
7.1.2.	Containers.....	99
7.1.2.1.	JBoss.....	100
7.1.2.2.	Apache Tomcat.....	101
7.1.3.	Apache Axis .....	101



7.1.4.	PostgreSQL.....	101
7.2.	Implementação do Sistema.....	102
7.2.1.	Modelos UCON <sub>ABC</sub> utilizados.....	102
7.2.1.1.	Utilização do Modelo UCON <sub>preA1</sub> .....	102
7.2.1.2.	Utilização do Modelo UCON <sub>onA2</sub> .....	103
7.2.1.3.	Utilização do Modelo UCON <sub>preB1</sub> .....	103
7.2.1.4.	Utilização do Modelo UCON <sub>onB2</sub> .....	103
7.2.1.5.	Utilização do Modelo UCON <sub>preC0</sub> .....	104
7.2.1.6.	Utilização do Modelo UCON <sub>onC0</sub> .....	104
7.2.2.	Banco de Dados.....	104
7.2.2.1.	Tabelas de Atributos dos Usuários.....	105
7.2.2.2.	Tabelas de Definição de AISs.....	108
7.2.3.	Configuração do Apache Axis.....	111
7.2.3.1.	Configuração dos Processos e do Controlador UCON <sub>ABC</sub> .....	112
7.2.3.2.	Falha do Apache Axis.....	114
7.2.4.	Autenticação e Geração do Objeto de Sessão.....	116
7.2.5.	Filtros.....	119
7.2.5.1.	Filtro de Autorização.....	119
7.2.5.2.	Filtro de Obrigação.....	121
7.2.5.3.	Filtro de Condição.....	121
7.2.6.	Páginas do Sistema.....	122
7.3.	Conclusão do Capítulo.....	126
8.1.	Introdução.....	127
8.2.	RF-B2B.....	127
8.2.1.	Funcionalidades.....	128
8.2.2.	Tecnologias Utilizadas.....	129
8.3.	Puma Motors.....	130

8.4.	Localização Física .....	131
8.5.	Controle de Acesso .....	132
8.5.1.	Gerenciamento de Permissões .....	132
8.5.2.	Funcionamento .....	133
8.6.	Conclusão do Capítulo .....	135
9.1.	Contribuições Deste Trabalho .....	136
9.2.	Trabalhos Futuros .....	138

- Processos de Segurança .....	24
- Matriz de Acesso .....	27
- Lista de Controle de Acesso (a) e Capacidade (b) .....	28
- Modelo Obrigatória .....	30
- Modelo RBAC .....	33
- Componentes de um Sistema DRM .....	38
- Controle de Acesso Tradicional .....	42
- Propriedades de Continuidade e Mutabilidade .....	43
- Componentes do Modelo UCON <sub>ABC</sub> .....	45
- Estrutura Conceitual do Monitor de Referência UCON .....	56
- Abrangência do UCON .....	58
- Funcionamento de um Sistema de CE B2B .....	61
- Camadas de Integração de Sistemas B2B .....	69
- Fluxo de funcionamento do <i>WebSphere Commerce Suíte</i> .....	73
- Tabelas que compõem as informações para o controle de acesso .....	74
- Diagrama de Componentes GAMMA .....	79
- Arquitetura do processador de controle de acesso distribuído.....	81
- Modelo de Implementação Proposto.....	85
- Agrupamento Implícito no Gerenciamento de Permissões.....	89
- Relação de permissões entre Sujeitos e Objetos através do RBAC.....	92
Relação de permissões entre Sujeitos e Objetos através do AI.....	92
- Relação de permissões entre Sujeitos e Objetos através do AIP.....	93
- E Cobertura do Modelo Proposto.....	95
- Arquitetura de Containers J2EE .....	100
- Tecnologias de Desenvolvimento.....	102
- DER das Tabelas de Atributos dos Usuários.....	105
- Esquema Lógico das Tabelas de Atributos dos Usuários.....	106

- DER das Tabelas de Definição dos AISs.....	108
- Esquema Lógico das Tabelas de Definição dos AISs.....	109
- Exemplo de Configuração de um Processo.....	112
- Arquivo WSDD de Configuração de um Processo.....	113
- Método de um Bloqueador.....	115
0 - Modelo de Implementação utilizando um Bloqueador.....	116
- Diagrama UML de Classes para o Objeto de Sessão.....	118
- Página do SEC para autenticação no SEP.....	123
- Página de termo de adesão.....	123
- Página inicial após autenticação no SEP.....	124
- Página de fornecimento de senha crítica.....	124
- Listagem de produtos do SEP.....	125
- Página de acesso não autorizado.....	125
Relação com parceiros comerciais.....	128
Tecnologias RF-B2B.....	129
Tecnologias Puma Motors.....	130
Localização Geográfica das Empresas.....	131
- Controle de Acesso entre Puma Motors e Rodas Forte.....	134

– Modelos UCON <sub>ABC</sub> .....	49
– Relação de permissões entre Sujeitos e Objetos.....	91
Relação de acesso entre Grupos e Serviços.....	132

- Access Control List
- Agrupamento Implícito
  - Agrupamento Implícito de Objetos
  - Agrupamento Implícito Parcial
  - Agrupamento Implícito de Sujeitos
- Application Programming Interface
- Business-to-Business
- Business-to-Consumer
- Communication Agreement
- Comércio Eletrônico
  - Central Processing Unit
    - Common Object Request Broker Architecture
  - Client-side Reference Monitor
- Control Set
  - Discretionary Access Control
    - Distributed Component Object Model
  - Diagrama de Entidade e Relacionamento
  - Digital Rights Management
  - Dynamic Separation of Duty
  - Embedded Control Architecture w/ Message Push
  - Embedded Control Architecture w/ External Repository
  - Electronic Data Interchange
  - Empresa Fabricante de Processadores
  - Enterprise Java Bean
    - Empresa Montadora de Computadores
  - Enterprise Resources Planning
    - Generic Authorization Mechanisms for Multi-Tier Applications

- Mandatory Access Control
- Módulo de Decisão de Uso
- Módulo de Execução de Uso
- Payment Based Type
- Payment Free Type
  - Role-Based Access Control
- Remote Method Invocation
- Remote Procedure Call
- Security Definition Language
- Sistema da Empresa Consumidora de Processos
- Sistema da Empresa Provedora de Processos
  - Simple Object Access Protocol
- Server-side Reference Monitor
- Static Separation of Duty
  - Usage Control
- Value-Added Network
- Virtual Machine
- Value Object
  - Extensible Access Control Markup Language
- eXtensible Markup Language
  - Web Service Deployment Descriptor
  - Web Services Description Language

Recentemente foi proposto um modelo de controle de acesso, denominado  $UCON_{ABC}$ , que além de unir alguns dos principais conceitos de controle de acesso ainda propõe novos conceitos como: obrigações, condições, continuidade e mutabilidade. Apesar de abrangente, o  $UCON_{ABC}$  possui limitações e existem ainda muitas melhorias a serem pesquisadas, como por exemplo, a definição de uma forma adequada da aplicação deste modelo em Sistemas de Comércio Eletrônico (CE) *Business-to-Business* (B2B). Publicações científicas nesta área afirmam que são necessárias pesquisas na especificação, validação e execução de políticas de controle de acesso para sistemas B2B. Esta dissertação possui como principal contribuição científica a proposta de uma forma de aplicação do  $UCON_{ABC}$  em sistemas de CE B2B que interajam entre si. Além disso, é proposto o Agrupamento Implícito Parcial, uma técnica que facilita o gerenciamento de permissões neste tipo de sistema. A aplicabilidade da proposta desta dissertação é apresentada através de uma descrição detalhada da implementação de um sistema de CE B2B onde o controle de acesso segue as especificações desta proposta. Por fim, é apresentado um estudo de caso em que é possível visualizar, através de um exemplo do mundo real, a aplicação da proposta desta dissertação neste tipo de sistema.



Recently a model of access control was proposed, denominated  $UCON_{ABC}$  that besides uniting some of the main concepts of access control still proposes new concepts such as: obligations, conditions, continuity and mutability. In spite of including,  $UCON_{ABC}$  possesses many limitations and many improvements need to be made; for instance, the definition of an appropriate form of the application of this model in Systems of Business-to-Business (B2B) Electronic Commerce (EC). Scientific Publications in this area affirms that they are necessary researches in the specification, validation and execution of politics of access control for systems B2B. This dissertation possesses as main scientific contribution the proposal of a form of application of  $UCON_{ABC}$  in systems of B2B EC that interact among it self. Besides, it is proposed the Partial Implicit Grouping, a technique that it facilitates the administration of permissions in this system type. The applicability of the proposal of this dissertation it is presented through a description detailed of the implementation of a system of B2B EC where the access control it follows the specifications of this proposal. Finally, a case study is presented in that it is possible to visualize, through an example of the real world, the application of the proposal of this dissertation in this system type.

Na última década, a Internet cresceu significativamente de forma a mudar o cotidiano das pessoas e organizações. Onde anteriormente apenas havia o interesse em compartilhar informações através de arquivos e páginas estáticas, com o surgimento de novas tecnologias, foi possível desenvolver sistemas de conteúdo dinâmico com suporte a transações financeiras. Para acompanhar tais mudanças, os estudos na área de controle de acesso também têm evoluído de forma significativa.

O controle de acesso tem por objetivo limitar as ações que o usuário de um sistema possa executar de forma a manter a integridade, confidencialidade e disponibilidade dos dados. Tradicionalmente, sistemas de controle de acesso visam controlar apenas processos executados internamente em uma empresa. Em sistemas de Comércio Eletrônico (CE) *Business-to-Business* (B2B) existe a necessidade de se estender este controle de acesso para fora das fronteiras de uma empresa, de forma que seus parceiros comerciais possam tanto ter acesso, como fornecer informações.

Esta dissertação apresenta uma proposta de aplicação do  $UCON_{ABC}$ , um modelo de controle de acesso surgido em 2002, em sistemas de CE B2B que interagem entre si. A proposta utiliza todas as características do  $UCON_{ABC}$  que podem ser aplicadas especificamente neste tipo de sistema.

Outra contribuição apresentada é o Agrupamento Implícito Parcial (AIP), uma variação simplificada do Agrupamento Implícito (AI). Seu objetivo é prover meios de gerenciar eficientemente permissões de acesso em um sistema de CE B2B, mas de uma forma mais simplificada que o AI.

Para fins de validação da proposta de aplicação, esta dissertação apresenta uma descrição detalhada da implementação do protótipo de um sistema de CE B2B que utiliza os conceitos desta proposta.

Com o objetivo de auxiliar a compreensão da aplicação do modelo  $UCON_{ABC}$  em sistemas de CE B2B, esta dissertação ainda apresenta um estudo de caso entre duas indústrias parceiras que possuem este tipo de sistema para a realização de negócios.

Controle de acesso é uma área que vem se desenvolvendo há anos. No entanto, atualmente ainda existem determinados tipos de sistemas que possuem características que exigem soluções específicas para o controle de acesso. (MEDJAHED et al, 2003), por exemplo, afirma que são necessárias pesquisas na especificação, validação e execução de políticas de controle de acesso para sistemas de CE B2B.

A interação entre sistemas de CE B2B tem despertado muita atenção de pesquisadores. Através da interação é possível que sistemas de CE B2B de diferentes tecnologias interajam entre si no intuito de agilizar o processo de negociação entre empresas parceiras. Trabalhos como os de (MEDJAHED et al, 2003), (DABOUS, RABHI & RAY, 2003) ou (QUIX, SCHOOP & JEUSFELD, 2002) enfatizam o estudo em técnicas, métodos ou tecnologias necessárias para a interação entre sistemas B2B. Com isto, é possível perceber que a interação neste tipo de sistema não é mais uma tendência, mas sim uma realidade.

A interação entre sistemas B2B possibilita que usuários externos acessem informações de uma empresa usando sistemas específicos. Os sistemas específicos das empresas, através de seus usuários, concretizam os negócios. Portanto, o controle de acesso não pode ser tratado de forma convencional nessas interações. Nestes casos, quem requisita o acesso ao sistema de uma empresa parceira é uma entidade “composta”, isto é, uma entidade formada por um usuário de um sistema, com seus atributos individuais, e pelo próprio sistema, com seus atributos. Portanto, é necessária a definição de maneiras eficientes de implementar o controle de acesso especificamente entre sistemas que interagem entre si.

Muitos dos trabalhos publicados na área de controle de acesso em sistemas de CE B2B como (GOODWIN, GOH & WU, 2002), (ESSMAYR, PROBST & WEIPPL, 2004) ou (ROBISON, 2002) não consideram a interação entre os sistemas dos parceiros envolvidos. Apesar de (KRAFT, 2002) considerar a interação, ele introduziu um modelo abstrato de controle de acesso específico apenas para *Web Services*.

(PARK & SANDHU, 2002) se juntaram em um esforço no sentido de unir alguns dos principais conceitos de controle de acesso e, além disso, propor melhorias. Como resultado, foi proposto um modelo chamado  $UCON_{ABC}$ . Devido à sua abrangência, o modelo

UCON<sub>ABC</sub> pode ser utilizado em diferentes sistemas e com arquiteturas variadas. (PARK, 2003) deu apenas o passo inicial para futuras pesquisas sobre o modelo UCON<sub>ABC</sub>. (PARK, 2003) afirma que existem ainda muitas melhorias a serem pesquisadas e desenvolvidas. Dentre elas, ele faz referência à pesquisa no UCON<sub>ABC</sub> em arquiteturas de segurança para Sistemas de CE B2B.

Estas pesquisas justificam a necessidade de se propor uma forma de aplicação do UCON<sub>ABC</sub> que atenda as necessidades específicas de sistemas de CE B2B que interagem entre si.

Este trabalho tem como objetivo geral a proposta de uma forma de aplicação do modelo UCON<sub>ABC</sub>, de maneira que atenda às necessidades de sistemas de CE B2B que interajam entre si, sem levar em consideração qualquer ferramenta de interação.

Para que o objetivo geral seja alcançado, os seguintes objetivos específicos podem ser enumerados:

- Definir uma forma de aplicação do modelo UCON<sub>ABC</sub> em sistemas de CE B2B.
- Definir quais dos modelos UCON<sub>ABC</sub> existentes podem ser aplicados no controle de acesso em sistemas de CE B2B.
- Desenvolver o protótipo de sistemas de CE B2B onde haja trocas de informações e negociação entre duas empresas parceiras através de uma ferramenta de interação.

Esta dissertação está organizada da seguinte maneira: No capítulo 2, como fundamentação para o estudo em controle de acesso, são abordados conceitos básicos

referentes a controle de acesso em sistemas computacionais. No capítulo 3 é apresentado o conceito fundamental para este trabalho: o modelo  $UCON_{ABC}$ , suas principais propriedades, arquiteturas e características. O capítulo 4 apresenta conceitos básicos sobre o Comércio Eletrônico *Business-to-Business*. No capítulo 5, é feito um estudo sobre as pesquisas relacionadas a controle de acesso em sistemas de CE B2B. O capítulo 6 apresenta a proposta de aplicação do  $UCON_{ABC}$  em sistemas de CE B2B. Nele é descrita a forma de funcionamento da proposta e, além disso, é feita uma comparação desta proposta com os trabalhos relacionados. No capítulo 7 são apresentadas as tecnologias utilizadas na implementação do protótipo do sistema, bem como a descrição do próprio sistema, que auxiliou na definição do modelo de implementação. Um estudo de caso que utiliza a proposta desta dissertação para desenvolvimento do controle de acesso de um sistema de CE B2B é apresentado no capítulo 8. Por fim, o capítulo 9 apresenta a conclusão desta dissertação, citando as contribuições científicas e propostas de trabalhos futuros.

Este capítulo aborda conceitos referentes a controle de acesso em sistemas computacionais. Primeiro são expostos os conceitos fundamentais de controle de acesso e de segurança. Posteriormente, serão abordados os três principais tipos de modelo de controle de acesso existentes. Por fim, serão apresentados novos conceitos de controle de acesso que estão atualmente em evidência.

Para que haja segurança em qualquer Sistema de Gerenciamento de Informação, é de fundamental importância a proteção de dados e recursos contra acessos não-autorizados no intuito de prover confidencialidade. Além disso, deve ser mantida a integridade dos dados contra modificações impróprias ou não-autorizadas ao mesmo tempo em que se deve assegurar a disponibilização de tais informações para os usuários legítimos (SAMARATI & VIMERATI, 2000). Tais requisitos de segurança são objetivos a serem alcançados através de uma área de pesquisa em segurança computacional denominada “Controle de Acesso”.

Nos últimos trinta anos, muitos esforços têm surgido com o objetivo de atingir um controle confiável no uso de recursos digitais. Apesar de ter havido avanços expressivos na área de controle de acesso, o conceito básico do modelo Matriz de Acesso tem permanecido imutável. Este modelo de controle de acesso foi proposto no início da década de 70 por Lampson e tem por principal característica mediar o acesso de um sujeito a um objeto. Muitas pesquisas na área propõem extensões e melhorias nos conceitos básicos da Matriz de Acesso (PARK & SANDHU, 2004).

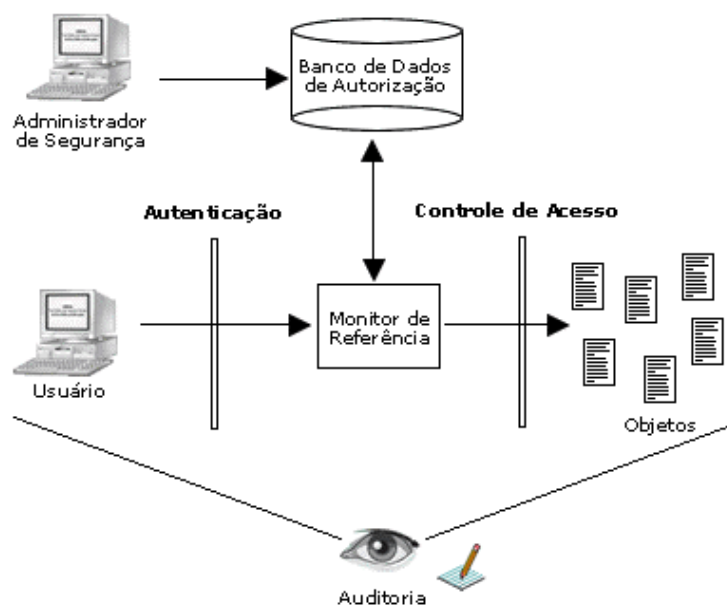
Segundo (SANDHU & SAMARATI, 1994), o objetivo do controle de acesso é limitar as ações que um usuário de um sistema computacional possa executar, impedindo atividades que possam levar a uma falha de segurança. Tal limitação é feita por um monitor de referência que intermedia todas as solicitações de usuários a recursos ou informações de

um sistema. Desta forma, o Monitor de Referência decide se a solicitação deve ser permitida ou negada.

Apenas o controle de acesso não garante um nível de segurança aceitável em um sistema computacional. Para (SANDHU & SAMARATI, 1996), a autenticação, o controle de acesso e a auditoria juntos provêm a base para a segurança de um sistema.

A autenticação tem por objetivo provar que uma entidade é realmente quem diz ser, como por exemplo: em um sistema computacional, um usuário deve provar, de alguma forma, que ele realmente é quem diz ser para que possa então ter acesso a processos ou informações restritas. A autenticação pode ser baseada em: i) alguma coisa que o usuário conhece, como uma senha; ii) alguma coisa que o usuário possui, como um *smart card*; iii) alguma coisa que o usuário é, exibido em algum sinal biométrico, como uma impressão digital ou a íris; iv) pode ser feita através de certificados digitais.

A auditoria tem como principal ferramenta os *logs*, ou seja, registros dos eventos do sistema. Os *logs* de auditoria são muito utilizados na administração e manutenção de sistemas em caso de falha de *software* ou *hardware*. Ela basicamente se preocupa na realização de uma análise posterior às ações dos usuários no sistema. Dentre as principais vantagens deste processo, podem ser citadas: i) Possibilidade de analisar o comportamento de usuários que utilizam o sistema a fim de encontrar possíveis tentativas de violação; ii) Possibilidade de determinar possíveis falhas na segurança do sistema; iii) Possibilidade de assegurar que o usuário não abuse de seus privilégios (SANDHU & SAMARATI, 1994). Apesar das vantagens existentes, (LANDWEHR, 2001) alerta pelo fato de haver a possibilidade do invasor do sistema apagar seus rastros através da destruição dos arquivos de *log*. Desta forma, é necessário que o processo de auditoria seja imune a ataques.



A Figura 2.1 ilustra a interação entre os três processos de segurança citados. É possível observar a existência de um Administrador de Segurança que gerencia as informações dos usuários do sistema em um Banco de Dados de Autorização. Este Banco de Dados fornece informações para que o Monitor de Referência possa, após a autenticação do usuário, realizar o controle de acesso aos objetos ou ações do sistema. Também é possível observar a presença de um processo de auditoria que monitora e armazena todas as ações realizadas no sistema.

A literatura científica relacionada a controle de acesso não estabelece um padrão sobre o que são políticas, modelos ou mecanismos de controle de acesso. (SANDHU & SAMARATI, 1994), por exemplo, classificaram os diferentes tipos de controle de acesso como políticas. Já (LANDWEHR, 1981) classificou-os como modelos. Tal indefinição causa uma certa confusão sobre qual termo correto se deve utilizar. Sendo assim, esta dissertação irá adotar os termos descritos nesta seção.



(GANTA, 1996) afirma que os *Modelos de Controle de Acesso* fornecem meios para que seja possível especificar, analisar e implementar as *Políticas de Controle de Acesso* em sistemas multi-usuários. As políticas são regras que têm por objetivo tornar um sistema computacional seguro. Estas regras devem estar sempre de acordo com o modelo no qual a política foi baseada. É necessário que os modelos sejam flexíveis o bastante para que sejam utilizados por vários tipos diferentes de políticas de segurança.

Já os *Mecanismos de Controle de Acesso* são os métodos, ferramentas ou procedimentos que são utilizados para assegurar qualquer política de controle de acesso. Estes mecanismos podem ser os certificados digitais, o protocolo SSL, métodos de autenticação, dentre outros.

(PARK & SANDHU, 2004) definem como “Modelos Tradicionais” os modelos clássicos e mais conhecidos pela comunidade científica, os quais são Modelo Discricionário, Modelo Obrigatório e Modelo Baseado em Papéis. Sendo assim, para fins de organização deste capítulo primeiramente serão aqui exibidos os Modelos Tradicionais. Posteriormente serão exibidos os conceitos modernos de controle de acesso.

Em 1985 o (DoD, 1985) estabeleceu dois modelos de Controle de Acesso: o primeiro é o Modelo Discricionário que, devido à sua flexibilidade, pode ser utilizado nos mais variados ambientes como: comerciais, industriais ou educacionais. Normalmente conhecido como DAC (*Discretionary Access Control*). O segundo é o Modelo Obrigatório, que é particularmente utilizado em ambientes militares e é conhecido como MAC (*Mandatory Access Control*). Este segundo modelo será descrito posteriormente ao modelo discricionário.

O Modelo Discricionário tem por finalidade gerenciar o acesso de usuários a um objeto baseando-se em identidades individuais ou grupais de usuários e objetos. Segundo (DoD, 1985), a base para este modelo é que um usuário tem a permissão para especificar os tipos

de acesso que outros usuários podem ter sobre os objetos sob seu controle. Assim, os mecanismos de execução baseados neste modelo devem: i) Permitir que usuários especifiquem e controlem o compartilhamento de seus objetos a outros usuários, grupos de usuários, ou ambos; ii) Prover meios para limitar a propagação dos direitos de acesso; iii) Prover meios para que os objetos sejam protegidos de acessos não-autorizados; iv) Especificar, para cada objeto, uma lista de indivíduos e uma lista de grupos com seus respectivos modos de acesso; v) Especificar, para cada objeto, uma lista de indivíduos e uma lista de grupos onde o acesso não deve ser permitido; vi) Prover meios para que o acesso a um objeto, por usuários que ainda não tenham permissão, seja apenas permitido por usuários autorizados (superusuários).

O acesso de um usuário a informações é feito com base na identidade do usuário e nas autorizações que especificam, para cada usuário e objeto do sistema, os modos de acesso (ler, escrever ou executar). Em cada solicitação de acesso que o usuário fizer a um objeto, as autorizações são verificadas. Caso o usuário possa acessar o objeto no modo desejado, o acesso é permitido, caso contrário é negado (SANDHU & SAMARATI, 1994).

O Modelo Discricionário tem por base os conceitos do primeiro modelo de controle de acesso proposto: a Matriz de Acesso.

A Matriz de Acesso é um modelo de controle de acesso proposto por (LAMPSON, 1971) que tem sido muito utilizado devido a sua simplicidade e generalidade, permitindo assim uma grande variedade de técnicas de implementação.

Este modelo possui três componentes principais: i) Os *Objetos* são entidades passivas compreendidas por: arquivos, diretórios ou programas em sistemas operacionais, ou então tabelas, visões ou procedimentos em sistemas de Banco de Dados. ii) Os *Sujeitos* são entidades ativas que executam atividades e solicitam acesso aos objetos. Eles podem ser: usuários, processos ou computadores. iii) Conjunto de *Regras* que gerenciam a manipulação dos Objetos pelos Sujeitos. Tendo por base estes três componentes, é possível compreender a definição de que a Matriz de Acesso é um modelo conceitual que especifica os direitos que cada sujeito pode ter sobre cada objeto através das regras.

Este modelo é representado por uma matriz com uma linha para cada Sujeito e uma coluna para cada Objeto. A intersecção entre uma linha e uma coluna contém o modo de acesso entre um sujeito e um objeto correspondente. O modo de acesso depende do sistema em questão. Levando em consideração sistemas operacionais, os modos podem ser: ler, escrever, anexar ou executar (LANDWEHR, 1981). A função de assegurar que apenas as operações autorizadas pela Matriz de Acesso sejam executadas, é atribuída ao Monitor de Referência. Ele é responsável por intermediar todas as tentativas de acesso dos sujeitos sobre os objetos.

Através da Figura 2.2, (SAMARATI & VIMERATI, 2000) ilustraram a forma de funcionamento da Matriz de Acesso. Nela estão representados direitos de ler (*Read*), escrever (*Write*) e executar (*Execute*). Existem quatro objetos, três arquivos e um programa, os quais os sujeitos Ana, Bob e Carlos têm acesso. É possível observar que Ana é a proprietária do Arquivo 1, tendo assim o direito de ler e escrever sobre o mesmo. No entanto, ela não tem acesso ao Arquivo 3. O proprietário de um arquivo tem a possibilidade de atribuir ou cancelar direitos de acesso de seus arquivos a outros usuários. Assim, de acordo com a figura, Ana atribuiu somente o direito de leitura do Arquivo 1 para Bob. Além disso, nota-se que Ana possui apenas o direito de execução sobre o Programa 1, enquanto que Carlos pode tanto lê-lo, quanto executá-lo.

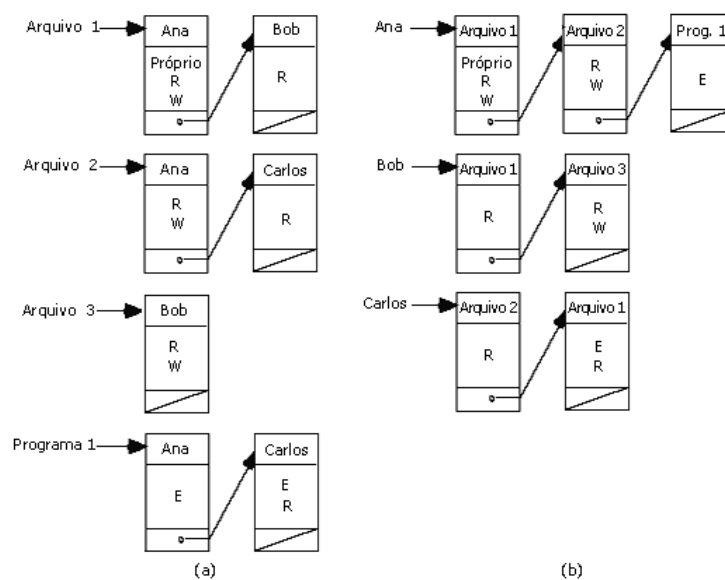
	Arquivo 1	Arquivo 2	Arquivo 3	Programa 1
Ana	Próprio R W	R W		E
Bob	R		R W	
Carlos		R		E R

Como se pode observar no exemplo da Figura 2.2, o modelo da Matriz de Acesso especifica apenas que existem regras (ler, escrever, ou executar), mas não especificam a semântica destas regras. É devido a esta liberdade de se impor diferentes regras para diferentes casos, que este modelo tem grande flexibilidade e aplicabilidade.

Apesar das vantagens existentes deste modelo, sua implementação é inviável em sistemas reais, pois ela seria muito grande em tamanho e esparsa pelo fato de a maioria das intersecções entre linhas e colunas estarem vazias. Assim, haveria um desperdício de memória. Para solucionar este problema, foram desenvolvidas duas diferentes técnicas para implementar a Matriz de Acesso de forma viável (SANDHU & SAMARATI, 1994).

A primeira técnica é a *Lista de Controle de Acesso* (ACL – *Access Control List*), representada pela Figura 2.3(a). Nela a matriz é armazenada em colunas, de forma que cada objeto seja associado com uma lista que indica os sujeitos que o acessam. Em cada nó da lista também são descritos os tipos de acesso que um determinado sujeito da lista pode ter ao objeto em questão. Esta técnica tornou simples tanto encontrar todos os sujeitos que acessam um determinado objeto, quanto negar acesso a um objeto, bastando apenas trocar uma lista existente por outra vazia.

Apesar das facilidades apresentadas, esta técnica possui limitações quando se deseja encontrar todos os objetos que um determinado sujeito tem acesso. Para isto, faz-se necessário uma busca percorrendo cada elemento nas listas de todos os objetos existentes no sistema. O mesmo procedimento deve ser feito para que se possa negar o acesso de um sujeito a todos os objetos os quais o mesmo tenha acesso.



Segundo (SAMARATI & VIMERATI, 2000), Sistemas Operacionais modernos normalmente utilizam a técnica das ACLs. Alguns implementam uma forma de ACL abreviada na qual são permitidas autorizações individuais. As autorizações são somente atribuídas para um número limitado de grupos de usuários, normalmente um ou dois. Em sistemas Unix, cada usuário no sistema pertence um grupo e cada arquivo tem um dono e é associado a um grupo, normalmente o de seu dono. Assim, a autorização para cada arquivo pode ser especificada pelo seu dono para o grupo o qual o arquivo pertence, ou para o resto do mundo.

A segunda técnica corresponde à *Capacidade (Capability)*, representada pela Figura 2.3(b). Nesta técnica, a matriz é armazenada por linha, de forma que cada usuário seja associado com uma lista de objetos a que o mesmo possui acesso. Esta lista denomina-se Lista de Capacidade. Em cada nó da lista também são descritos os tipos de acesso que o sujeito pode ter a cada objeto. Com esta técnica, é fácil encontrar todos os objetos os quais um determinado sujeito possui acesso, bastando apenas examinar a Lista de Capacidade do sujeito.

Esta técnica também possui limitações. De forma similar à ACL, quando se deseja encontrar todos os sujeitos que possuem acesso a um determinado objeto. Para isto, faz-se necessário uma busca percorrendo cada elemento nas listas de todos os sujeitos cadastrados no sistema. Segundo (SANDHU & SAMARATI, 1994), na década de 70 foram desenvolvidos vários sistemas computacionais baseados em capacidade. No entanto, estes sistemas não obtiveram sucesso comercialmente.

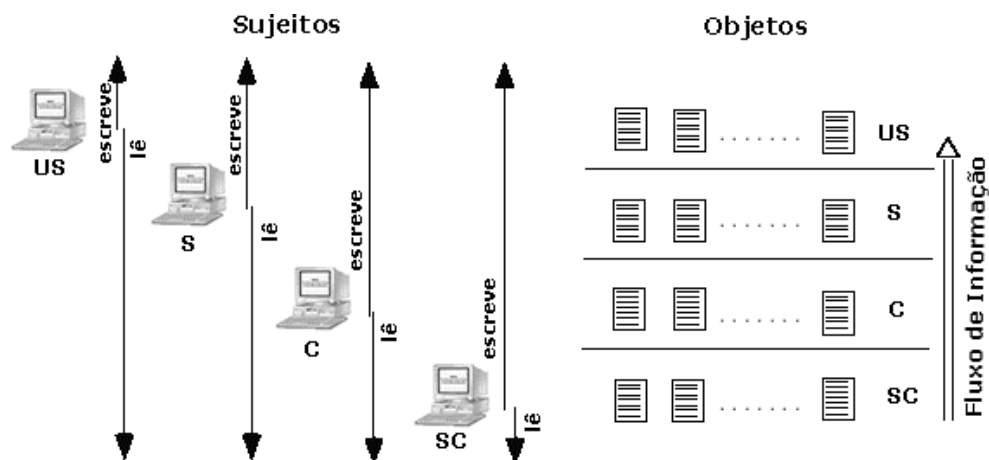
O Modelo Obrigatório possui Sujeitos e Objetos que são classificados baseando-se em níveis de segurança pré-definidos, os quais são utilizados no processo de decisão de acesso. O nível de segurança associado a um Objeto depende da importância de seu conteúdo, em que dados confidenciais e de alto valor são classificados em um alto nível. O nível de segurança associado a um Sujeito, depende da competência e grau de confiança que ele possui dentro de uma organização. Em sistemas militares, normalmente a hierarquia de

níveis adotados segue a seguinte ordem: Ultra Secreto (US), Secreto (S), Confidencial (C) e Sem Classificação (SC), onde  $US > S > C > SC$  (SANDHU & SAMARATI, 1994).

Um usuário que se conecta em um sistema em uma determinada classe de acesso, origina um Sujeito correspondente àquela classe. As solicitações que um Sujeito faz para acessar um Objeto são controladas através das classes de acesso do Sujeito e do Objeto. Estas solicitações são concedidas apenas se as condições, referente às classes, estiverem de acordo com os seguintes princípios:

- *Não-ler-acima (No-read-up)*: é permitido o acesso de leitura de um sujeito a um objeto apenas se a classe de acesso do sujeito for dominante em relação à classe de acesso do objeto.
- *Não-escrever-abaxo (No-write-down)*: é permitido o acesso de escrita de um sujeito a um objeto apenas se a classe de acesso do objeto for dominante em relação à classe de acesso do sujeito.

Através da Figura 2.4, é possível observar que, de acordo com os dois princípios citados, um Usuário que se conecta ao sistema em uma classe “Ultra-Secreta”, o Sujeito criado possui o nível mais alto na hierarquia e pode ter acesso de leitura a todos os outros Objetos das classes inferiores. À medida que um Usuário vai se conectando em níveis mais baixos, seus direitos de leitura diminuem e de escrita aumentam. Tal característica possibilita um maior controle sobre o segredo das informações num sistema.



É possível que um Usuário se conecte no sistema em diferentes classes de acesso, desde que sejam menores que a sua classe original. Isto devido ao princípio de “não-escrever-abaixo”. Assim, um usuário pertencente à classe Secreta (S) pode apenas ter acesso de “escrita” em um objeto da classe Confidencial (C) se este usuário conectar-se na classe Confidencial ou inferior.

Com a popularização da computação no início da década de 90, diversas empresas e governos civis despertaram interesse em desenvolver sistemas de processamento de informação de grande porte que atendessem às suas necessidades financeiras ou operacionais. Estes sistemas manipulariam informações confidenciais e de alto valor. O acesso dessas informações por pessoas não-autorizadas, a alteração indevida ou o seu roubo, poderia causar sérios danos legais, financeiros ou de privacidade pessoal. Pesquisadores na área de segurança perceberam que os modelos de controle de acesso existentes não atendiam a todas as necessidades que estes novos sistemas exigiam. Diversas alternativas de modelos de controle de acesso foram propostos para tentar prover tais necessidades. Dentre eles, surgiu o Modelo de Controle de Acesso Baseado em Papel, mais conhecida como Modelo RBAC (*Role-Based Access Control*).

A primeira vez em que se ouviu falar do RBAC foi em 1992 com a publicação de (FERRAILOLO & KUHN, 1992). Em 1996, (SANDHU, 1996) definiu uma família de modelos que ficou conhecida como RBAC96. Antes da definição desta família, não havia uma divisão no modelo RBAC, fazendo com que ou ele fosse muito complexo para determinados casos ou demasiadamente simples para outros. Com a família RBAC96, foi possível determinar um modelo correto para cada situação em particular.

Nos últimos anos, produtos como Bancos de Dados e Sistemas Operacionais vêm utilizando o RBAC como modelo de controle de acesso. Além disso, diversos modelos novos vinham sendo propostos sem se preocupar com a padronização do RBAC. O *National Institute of Standards and Technology* (NIST), uma agência do Departamento de Comércio de Administração Tecnológica dos Estados Unidos, tem feito esforços para

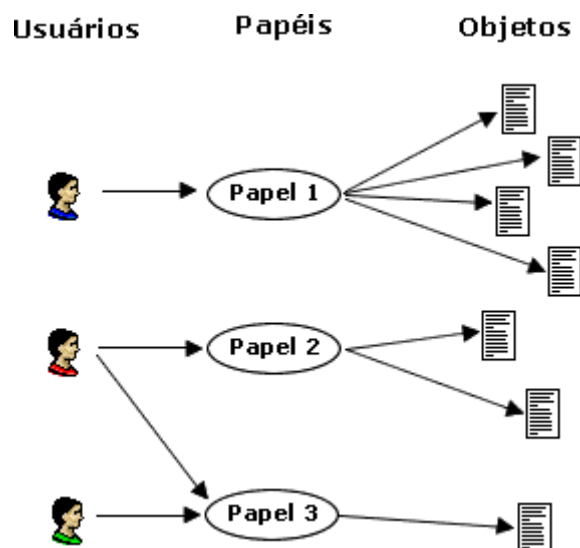
resolver este problema unificando idéias de modelos citados pela comunidade científica e modelos utilizados em produtos comerciais. Desta forma, em 2001 (FERRAILOLO et ALL, 2001) definiram quatro modelos padrão para RBAC: Modelo RBAC Básico, Modelo RBAC Hierárquico, Separação Estática de Deveres e Separação Dinâmica de Deveres. A descrição destes modelos será feita na seção 2.4.2.

Na publicação que deu origem ao Modelo RBAC, (FERRAILOLO & KUHN, 1992) se ativeram ao fato de que em muitas organizações, os usuários finais não eram os donos das informações pelas quais os mesmos tinham direito de acesso. Para estas organizações, elas próprias eram as proprietárias tanto das informações, quanto dos programas que processavam estas informações. Assim, eles afirmaram que o problema das políticas de controle de acesso existentes até então era que elas faziam o controle baseando-se nas propriedades dos Objetos e Sujeitos ao invés de se basearem nas funções dos empregados.

Segundo (FERRAILOLO & KUHN, 1992), normalmente as decisões de controle de acesso baseiam-se em papéis que são determinados de acordo com os deveres, responsabilidades ou qualificações que um usuário possui em uma organização. Assim, as decisões do RBAC baseiam-se nas funções as quais um usuário é autorizado a executar em uma organização. Como exemplo de papéis em uma organização, pode-se citar um sistema bancário no qual pode haver papéis como: caixa, contador ou gerente.

O Modelo RBAC exige a definição prévia de papéis no sistema. Estes papéis podem ser definidos como um conjunto de ações e responsabilidades associadas com uma determinada atividade de trabalho. Desta forma, ao invés de especificar todas as ações ou objetos que cada usuário tem o direito de executar ou acessar, o RBAC permite que as autorizações de acesso a objetos sejam especificadas e associadas a papéis (SANDHU & SAMARATI, 1994). A Figura 2.5 ilustra o relacionamento entre usuários, papéis e objetos. Um usuário, possuidor de um papel, pode ter acesso a todos os objetos que seu papel lhe permite. Dependendo do sistema, um usuário pode fazer uso de papéis diferentes em ocasiões diferentes ou até mesmo simultaneamente. Da mesma forma, um papel pode ser atribuído a vários usuários.





A padronização do Modelo RBAC foi proposta por (FERRAILO et ALL, 2001). Esta seção tem por finalidade ilustrar, de forma básica, os modelos propostos.

O primeiro modelo é o *RBAC Básico*, que é composto por um conjunto de cinco elementos básicos, denominados: usuários (USERS), papéis (ROLES), objetos (OBS), operações (OPS) e permissões (PRMS). Este modelo é definido fundamentalmente em termos de usuários estarem associados a papéis e permissões estarem associadas a papéis, permitindo assim um relacionamento de muitos – para - muitos entre usuários e permissões. Além disso, este modelo possui um conjunto de sessões (SESSIONS), que são mapeamentos entre usuários e conjuntos de papéis.

O modelo *RBAC Hierárquico* introduz ao modelo RBAC Básico o conceito de hierarquia de papéis. As hierarquias são uma forma natural de estruturar papéis que reflitam linhas organizacionais de autoridade e responsabilidade. As hierarquias de papéis definem uma relação de herança entre papéis, em que um papel p1 pode herdar um papel p2, de forma que todos os privilégios de p2 sejam também privilégios de p1.

Na Política RBAC podem surgir conflitos de interesse como resultado da atribuição de papéis conflitantes a um usuário. O modelo de *Separação Estática de Deveres* (SSD – *Static Separation of Duty*) é uma maneira eficiente de prevenir esta forma de conflito, pois ele reduz o número de permissões que podem ser atribuídas a um usuário através da imposição de restrições em usuários que podem ser associados a vários papéis.

Por fim, o modelo de *Separação Dinâmica de Deveres* (DSD – *Dynamic Separation of Duty*) é similar ao estático, pois ele também tem por objetivo limitar as permissões que podem ser associadas a um usuário. No entanto, estes dois modelos se diferenciam pela forma como estas limitações são impostas. A separação dinâmica limita a disponibilidade de permissões aos usuários através de restrições impostas em papéis que podem ser ativadas nas sessões dos usuários. As propriedades DSD fornecem suporte para o princípio de menor privilégio onde, dependendo do papel que estiver sendo executado, cada usuário possui diferentes níveis de permissão em ocasiões diferentes. Estas propriedades asseguram que as permissões não persistam além do tempo necessário para a execução do dever.

O Modelo RBAC obteve tanta aceitação pela comunidade científica e empresarial que (SANDHU, 2001) chegou a dizer que este foi o modelo de controle de acesso dominante na década de 90 e, além disso, fez a previsão de que este modelo continuaria dominante nesta década. A aplicação do modelo RBAC em sistemas de Comércio Eletrônico *Business-to-Business* (B2B) e *Business-to-Consumer* (B2C) seria o principal motivo da predominância do RBAC nesta década. Isto se deve ao fato de este modelo possuir meios de separar responsabilidades em sistemas entre organizações. Sua aplicação na economia digital poderia permitir casos em que papéis fossem adquiridos através de pagamentos, ou então comercializados entre usuários.

Nos últimos anos, novos conceitos vêm sendo propostos pela comunidade científica para suprir novas necessidades provenientes de aplicações voltadas para Web, mais

especificamente para o Comércio Eletrônico. Destes novos conceitos, dois se destacam: Gerenciamento de Confiança (*Trust Management*) e Gerenciamento de Direitos Digitais (DRM - *Digital Rights Management*).

O Gerenciamento de Confiança foi introduzido por (BLAZE et All, 1996) como uma técnica unificada para especificar e interpretar políticas de segurança, credenciais e relacionamentos de confiança que permitam uma autorização direta de ações de segurança crítica. Para compreender de forma clara a relação entre estes três elementos, é possível tomar como exemplo um sistema de banco eletrônico. A *política de segurança* deste banco especifica que, para que possa ser feito um empréstimo de R\$1.000.000,00, é necessário que haja a aprovação de pelo menos  $n$  gerentes do banco. Uma *credencial* pode ser compreendida como um meio de um gerente provar ao sistema de que ele é um dos  $n$  gerentes que fazem a aprovação do empréstimo. As credenciais são definidas pelas chaves-públicas de usuários acrescidas de suas atribuições de confiança específicas, possibilitando gerar as autorizações necessárias. Os *relacionamentos de confiança* correspondem ao fato de o sistema do banco possibilitar a especificação de quais gerentes do banco podem assumir as credenciais para aprovação de empréstimo.

O Gerenciamento de Confiança baseia-se nos seguintes princípios: i) *Mecanismo Unificado*: políticas, credenciais e relacionamentos de confiança devem ser expressos em uma linguagem de programação segura para que as aplicações controlem a segurança de uma forma compreensiva, consistente e transparente. ii) *Flexibilidade*: o sistema de gerenciamento de confiança deve ser flexível o bastante para suportar relacionamentos de confiança complexos que podem ocorrer em aplicações de rede de grande porte. Da mesma forma, ele deve ser capaz de compreender políticas, credenciais e relacionamentos que sejam simples. iii) *Controle Local*: cada parte da rede pode decidir em cada circunstância se aceita as credenciais apresentadas por uma segunda parte. iv) *Separação entre a política e o mecanismo*: o mecanismo para a verificação de credenciais não pode depender das credenciais ou da semântica das aplicações que as utiliza. Desta forma, é possível que

diferentes aplicações com diferentes políticas possam compartilhar a mesma infraestrutura de verificação (BLAZE et All, 1996).

Sistemas de controle de acesso tradicionais processam a autorizações através da combinação entre autenticação e controle de acesso. Assim, o sistema primeiro determina quem fez uma determinada solicitação de acesso para então decidir, através de uma consulta interna ao Banco de Dados, se deve permitir o acesso. Segundo (BLAZE et All, 1999), esta é uma técnica inadequada para um ambiente interconectado e dinâmico como a Internet. Eles afirmam que em um ambiente distribuído existe uma enorme quantidade de pessoas que podem fazer solicitações de acesso. Os grupos de pessoas que fazem estas solicitações estão em constante mudança e não podem ser conhecidos com antecedência. Além disso, saber quem fez uma solicitação se torna ainda mais complicado para as pessoas que fizeram pela primeira vez esta solicitação ao sistema. É necessário que haja uma maior flexibilidade e uma técnica para autorização distribuída. O correto, neste caso, seria saber se a chave que assinou esta solicitação está autorizada a executar uma determinada ação.

Os sistemas de Gerenciamento de Confiança autorizam o acesso a ações críticas baseando-se na resposta da seguinte pergunta: O conjunto de *credenciais*  $C$  prova que a *solicitação*  $S$  está de acordo com a *política de segurança local*  $P$ ?

Cada entidade que recebe solicitações deve ter uma política que serve como uma fonte de autoridade. Tal política poderá autorizar que certas ações sejam atribuídas a emissores de credencial que tem o poder de distribuir credenciais a usuários de confiança. A máquina de gerenciamento de confiança é um componente separado do sistema que recebe  $(C, S, P)$  como entrada. Avaliando estes dados, a máquina a resposta permitindo ou não o acesso (BLAZE et All, 1999).

Devido à evolução da internet e ao surgimento de novas tecnologias digitais, tornou-se possível criar e distribuir diversas formas de conteúdos e informações em formatos digitais, como por exemplo: arquivos de imagens, áudio, vídeo, ou *software*. Assim, qualquer pessoa é capaz de fazer várias cópias destes arquivos e os distribuir indiscriminadamente e ilegalmente para vários outros usuários (DUHL & KEVORKIAN, 2001).

Para resolver este problema, uma nova tecnologia vem sendo desenvolvida para proteger o comércio, a privacidade, ou a propriedade intelectual de arquivos digitais. Esta tecnologia é denominada Gerenciamento de Direitos Digitais (DRM – *Digital Rights Management*). Segundo (KOENEN, LACY & MACKAY 2004), DRM é um conjunto de tecnologias que possibilitam assegurar a licença de informações digitais. Através de seu uso, torna-se possível distribuir eletronicamente conteúdos de valor, sem infringir os direitos autorais ou comerciais de seus proprietários. Além disso, DRM também pode ser utilizado para possibilitar a distribuição segura de conteúdos digitais, como por exemplo, a administração de documentos entre empresas ou então a manipulação de informações médicas de pacientes.

Um sistema DRM ideal deve possuir três características fundamentais. A primeira delas é o *Controle de Acesso*. Diferentemente dos sistemas convencionais de distribuição de dados, nos quais os dados são protegidos através de criptografia durante a transmissão, os sistemas DRM implementam o controle de acesso através do uso de métodos de linguagem de programação executados em um ambiente seguro. A segunda característica é a *Associação Segura entre Regras de Uso e Conteúdos Digitais*. Os sistemas DRM devem associar regras com conteúdos digitais. Estas regras determinam o uso de um conteúdo durante seu ciclo de vida. Por fim, há a característica de *Proteção Persistente*, onde os sistemas DRM devem ser projetados de forma proteger e administrar uma informação de forma persistente durante todo o seu ciclo de vida (KOENEN, LACY & MACKAY 2004).

Apesar de existirem várias formas diferentes de se implementar um sistema DRM, seu processo básico de funcionamento é sempre o mesmo. Este processo envolve quatro partes: o Provedor de Conteúdo, o Distribuidor, o Consumidor e a Casa de Controle. O sistema, representado pela Figura 2.8, trata-se de um Comércio Eletrônico que controla o fluxo de informações e de dinheiro entre as partes envolvidas (LIU et. All, 2003).



Neste sistema, o *Provedor de Conteúdo* tem por função tanto prover, quanto proteger os direitos digitais de conteúdos digitais como arquivos de filme, música, ou imagens. O *Distribuidor* age como se fosse uma loja de varejo *on-line*. Ele recebe conteúdos digitais de Provedores de Conteúdo e os distribui através da criação de um catálogo *web*. O *Consumidor* utiliza o sistema DRM para consumir um determinado conteúdo digital, adquirindo-o a partir de um Distribuidor, para então poder pagar por sua licença digital. Este conteúdo é utilizado através de uma aplicação específica que lhe possibilita solicitar para a Casa de Controle uma licença de uso mediante seu pagamento. A última parte envolvida é a *Casa de Controle*. Ela tem por objetivo controlar transações financeiras que envolvem a emissão de licenças digitais para os Consumidores mediante o pagamento de taxas de direitos autorais para o Provedor de Conteúdo e de taxas de distribuição para o Distribuidor.

De forma mais detalhada, primeiramente o Provedor de Conteúdo codifica o conteúdo digital em um formato compatível com o sistema DRM, pois diferentes sistemas DRM podem exigir formatos diferentes. Posteriormente, o Provedor de Conteúdo além de cifrar e empacotar o conteúdo digital para a sua distribuição, ainda pode introduzir uma marca d'água para que possa ser identificado tanto o proprietário quanto às regras de uso deste

conteúdo. As regras de uso especificam como um conteúdo deve ser usado. Segundo (DUHL & KEVORKIAN, 2001), estas regras de uso podem ser definidas por critérios como: i) *Preço*: quanto deve ser pago por um conjunto de direitos para um conteúdo, por exemplo; ii) *Duração*: licença para utilizar um conteúdo, por um ano, um mês ou uma semana, por exemplo; iii) *Frequência de Acesso*: é possível, por exemplo, imprimir dez vezes, ou escutar uma música três vezes; iv) *Redistribuição*: é possível que um conteúdo seja copiado, armazenado ou gravado em um CD por três vezes; v) *Transferência*: é possível que um conteúdo seja transferido para outro usuário ou dispositivo.

Posteriormente, o conteúdo é transferido para um servidor de Distribuição de conteúdo para que o mesmo seja distribuído *on-line*. Ao mesmo tempo, uma licença digital correspondente, contendo chaves de decifragem e regras de uso, é enviada para a Casa de Controle.

Para obter um conteúdo digital, um Usuário deve fazer um *download* deste conteúdo a partir de um servidor Web. Para fazer uso deste conteúdo, é necessário solicitar uma licença para a Casa de Controle. Depois que a Casa de Controle recebe uma solicitação de licença, ela verifica a identidade do Usuário solicitante, cobra um valor referente às regras de uso do conteúdo, gera relatórios de transação para o provedor de conteúdo e, por fim, a licença é enviada para a aplicação do Usuário depois que ele tiver pago através de um sistema de comércio eletrônico. Em posse da licença, a aplicação pode decifrar o conteúdo protegido e utilizá-lo de acordo com as regras de uso contidas na licença (LIU et. All, 2003).

Os sistemas DRM permitem que usuários redistribuam os conteúdos adquiridos para outros usuários. Esta prática, denominada super-distribuição, possibilita que os conteúdos sejam distribuídos para um grande número de compradores em potencial sem que haja um envolvimento direto de um Distribuidor. Apesar de haver esta liberdade, para fazer uso destes conteúdos, os Usuários que os receberem devem solicitar à Casa de Controle uma licença de uso e então pagar para o seu recebimento.

Este capítulo foi uma referência geral sobre os principais modelos de controle de acesso existentes, incluindo novos conceitos de controle de acesso. Esta visão geral sobre controle de acesso é de fundamental importância, pois, no Capítulo 3, estes assuntos serão abordados pelo fato do UCON herdar determinadas características destes conceitos.



Este capítulo ilustra os conceitos referentes a *Usage Control* (UCON). Muitos dos conceitos vistos no Capítulo 2 serão aqui abordados como parte da definição do UCON.

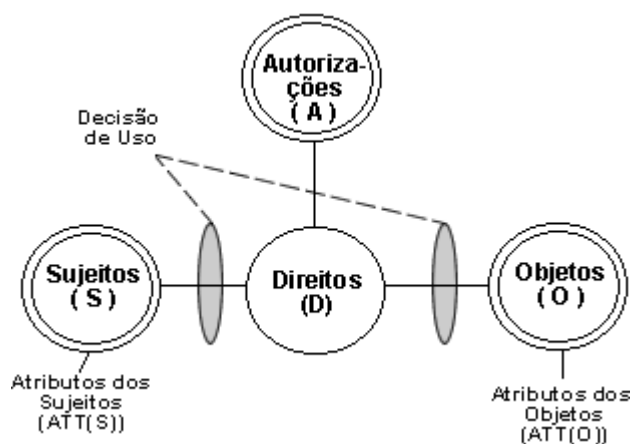
*Usage Control* é um novo conceito de controle de acesso introduzido por (PARK & SANDHU, 2002) cujo objetivo é fornecer um novo fundamento para o controle de acesso. Como o Controle de Acesso Tradicional, Gerenciamento de Confiança e DRM tentam resolver seus próprios problemas com soluções específicas para os mesmos, Park e Sandhu sentiram a necessidade de definir uma forma sistemática para o controle de acesso a objetos digitais, independentemente de circunstâncias específicas. Desta forma, o UCON foi definido como uma nova estrutura conceitual que abrange conceitos de Controle de Acesso Tradicional, DRM e Gerenciamento de Confiança de maneira sistemática para a proteção de recursos digitais.

Segundo (PARK & SANDHU, 2004), UCON não é um substituto para estes conceitos, pois além da junção destes, ele ainda abrange novos conceitos como: obrigações, condições, continuidade e mutabilidade, descritos a seguir.

Segundo (SANDHU & PARK, 2003), uma característica comum aos modelos de controle de acesso tradicionais, é que o processo de decisão de autorização é baseado em atributos de Sujeitos e Objetos, além dos direitos solicitados. A Figura 3.1 ilustra este controle de acesso baseado em atributos.

Apesar dos modelos de controle de acesso tradicionais atenderem às necessidades de muitas aplicações, atualmente sistemas de informação digital precisam de mais recursos do que simples autorizações. Em determinados momentos certas ações devem ser executadas por um Sujeito para possibilitar uma solicitação de uso. Desta forma, a decisão de uso deve

ser baseada no cumprimento de alguma ação, e não apenas nos atributos dos Sujeitos ou Objetos. No UCON, este fator de decisão é chamado de *oBrigação* e, juntamente com o processo de autorização, pode ser solicitado em aplicações modernas de controle de acesso. Um exemplo da aplicação desta característica está em um sistema de comércio eletrônico onde, para entrar no sistema, além de se autenticar o usuário deve obrigatoriamente ler e aceitar os termos de compromisso do sistema (SANDHU & PARK, 2003).



Existem certas situações que o acesso precisa ser limitado devido a certas condições do sistema ou do ambiente. Assim, o sistema deve verificar a situação atual do próprio sistema ou do ambiente para tomar as decisões de uso. No UCON, este fator de decisão chama-se *Condição* e pode ser exigido em aplicações modernas juntamente com a autorização e a obrigação. Esta característica permite com que o uso de certos recursos digitais no sistema possam ser permitidos apenas em alguns locais ou então em algumas horas do dia.

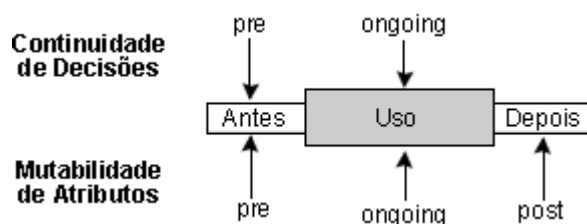
(PARK & SANDHU, 2002) alegam que a necessidade de obrigações e condições tem sido reconhecida por sistemas modernos de negócio, como por exemplo sistemas de comércio eletrônico B2C, ou então transações e interações B2B entre parceiros de negócios.

Em sistemas de controle de acesso tradicionais, o processo de autorização é feito antes que o acesso seja permitido. No entanto, sistemas de informação modernos exigem que este

processo de autorização seja estendido, avaliando as solicitações de uso durante o acesso do usuário no sistema. Esta propriedade é chamada de *Continuidade* e deve ser implementada em sistemas de controle de acesso modernos para o controle de uso relativamente longo ou para negação imediata de uso.

Outra característica de sistemas de controle de tradicionais é o fato de os atributos, tanto dos Sujeitos quanto dos Objetos, serem mutáveis apenas por ações administrativas. Atualmente, sistemas modernos exigem que tais atributos sejam modificados em consequência de efeitos colaterais das ações dos Sujeitos. Desta forma, as atualizações nos atributos podem ser feitas antes (*pre*), durante (*ongoing*), ou até mesmo depois (*post*) do processo de autorização. Políticas de segurança que exigem limites no número de acessos por um sujeito, ou então que reduzem o crédito de uma conta baseada em acesso pode ser facilmente especificada utilizando atributos mutáveis. Esta propriedade é chamada de

. Esta é uma importante inovação do UCON, pois ela o diferencia da maioria das propostas de melhorias de modelos de controle de acesso. Muitos sistemas de Comércio Eletrônico B2B ou B2C exigem alguma forma de mutabilidade em atributos. Soluções recentes de DRM são um exemplo disto. Créditos pré-pagos devem ser reduzidos de acordo com o tempo que o usuário exerce seus direitos sobre um determinado objeto. A Figura 3.2 ilustra tanto a propriedade de Continuidade, quanto a de Mutabilidade.



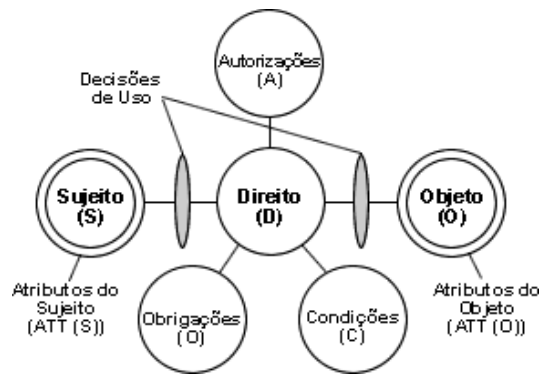
Segundo (SANDHU & PARK, 2003), apesar de algumas destas características já terem sido discutidas na literatura de controle de acesso, os focos são limitados a resolver problemas específicos, fazendo com que a discussão sobre o assunto não seja ampla. Desta forma, o UCON surgiu no intuito de abranger diversos modelos e propriedades de controle de acesso em uma única estrutura que supere estas faltas. No UCON, o controle de acesso

tradicional pode ser evoluído de forma que inclua controle de acesso moderno integrando obrigações e condições, assim como as autorizações que façam uso de continuidade e mutabilidade.

ABC foi o primeiro modelo que fez referência a uma extensão sistemática e compreensiva da Matriz de Acesso clássica. Ele integra Autorizações, oBrigações e Condições, juntamente com atributos mutáveis e execução contínua, em uma estrutura unificada. O modelo ABC é reconhecido como um modelo básico para o UCON, pois discute principalmente assuntos básicos de controle de acesso de Sujeitos sobre Objetos e não abrange qualquer assunto sobre o relacionamento entre diferentes Sujeitos e nem assuntos administrativos. Devido a esta estreita ligação, (PARK & SANDHU, 2004) renomearam o modelo ABC como  $UCON_{ABC}$ . Eles acreditam que este modelo será a base para a próxima geração de modelos de controle de acesso que serão utilizados por sistemas de segurança.

O processo de tomada de decisão em sistemas de controle de acesso tradicionais é feito através de autorizações. No modelo  $UCON_{ABC}$  o processo de tomada de decisão faz uso de atributos dos Sujeitos e Objetos. Além disso, ele ainda inclui oBrigações e Condições como parte do processo, o que possibilita uma tomada de decisão melhor e mais rica.

O modelo  $UCON_{ABC}$  consiste de oito elementos básicos: sujeitos, atributos do sujeito, objetos, atributos do objeto, direitos, autorizações, obrigações e condições. As autorizações, obrigações e condições são predicados funcionais que devem ser avaliados para a decisão de uso. Sujeitos com seus atributos, objetos com seus atributos e direitos podem ser divididos em vários componentes detalhados com perspectivas diferentes. A Figura 3.3 ilustra estes componentes (PARK & SANDHU, 2004).



Um sujeito é uma entidade que está associada a atributos, e que exerce certos direitos sobre os objetos. Um usuário de um sistema computacional pode, por exemplo, ser considerado um sujeito. No UCON, os Sujeitos subdividem-se em: i) *Sujeitos Consumidores (CS)*, que são entidades que exercem os direitos de acessar objetos. Exemplos deste tipo de sujeitos são os aparelhos tocadores de MP3, dispositivos de leitura de livros digitais, ou então usuários de uma intranet. ii) *Sujeitos Provedores (PS)* são entidades que fornecem um objeto e possuem certos direitos sobre ele. Exemplos para este tipo de entidade podem ser visualizados através na Figura 2.6 da seção 2.3.2. Nela existe o Provedor de Conteúdo, que tem por função prover e proteger os direitos digitais de conteúdos digitais, e o Distribuidor, que recebe e distribui objetos digitais. Estes dois elementos da arquitetura DRM podem ser tidos como exemplo de sujeitos provedores. iii) *Sujeito Identificador (IS)*, é uma entidade que é identificada por objetos digitais que possuem informações privadas. Um exemplo deste tipo de sujeito é um paciente com informações confidenciais de seu estado de saúde em um sistema de cuidados médicos.

Um sujeito é definido e representado por seus atributos. Desta forma, os atributos do sujeito são propriedades e capacidades de um sujeito que podem ser utilizados em um processo de decisão de acesso. Identidades, grupos, papéis, classes de acesso ou credenciais são exemplos deste tipo de atributo. Estes atributos podem ser tanto mutáveis quanto

imutáveis. Modelos de controle de acesso como Gerenciamento de Confiança ou DRM freqüentemente fazem uso de atributos mutáveis. Já modelos de controle de acesso tradicionais utilizam atributos imutáveis.

Os objetos são um conjunto de entidades sobre os quais sujeitos possuem direito de acesso ou uso. No  $UCON_{ABC}$  existem objetos do tipo *Originais* ou *Derivados*. Os objetos originais são arquivos simples de áudio, vídeo, imagem ou texto os quais um usuário pode ter direito de acesso. Os objetos derivados no UCON possuem uma definição diferente quando comparado com o DRM, em que eles são como objetos reproduzidos ou reusados. No  $UCON_{ABC}$  os objetos derivados são criados por consequência da obtenção ou execução de direitos sobre um objeto original. Um exemplo de objeto derivado é o arquivo de log criado através da execução de arquivos de música MP3. Estes objetos possuem propriedades UCON que são utilizadas para prover proteção mútua dos direitos de todos os sujeitos envolvidos: consumidor, provedor e identificador.

Da mesma forma que os sujeitos, os objetos também são associados a atributos que podem ser usados para as decisões de acesso. Exemplos destes atributos podem ser: níveis, permissões de papel, ou horários de uso. Os níveis classificam os objetos em uma hierarquia na Política de Controle de Acesso Obrigatório para que as autorizações possam ser feitas de acordo como nível do objeto (atributo do objeto) e nível do sujeito (atributo do sujeito). Valores podem ser utilizados para definir quantos créditos são necessários para obter direitos a um determinado objeto. Os atributos dos objetos também podem ser mutáveis.

Em modelos de controle de acesso tradicionais, direitos são definidos como privilégios que um sujeito pode ter e exercer sobre um determinado objeto em modos distintos, como por exemplo: ler, escrever ou executar. No modelo  $UCON_{ABC}$  este conceito possui uma

pequena diferença. Isto se deve ao fato de o modelo  $UCON_{ABC}$  levar em consideração as atividades do sujeito mesmo após sua autenticação e não apenas autorizar o acesso baseando-se nos atributos do sujeito ou objeto. A existência do direito no  $UCON_{ABC}$  é determinada quando o acesso é solicitado pelo sujeito. Assim, as funções de decisão de uso decidem se o acesso pode ou não ser permitido baseando-se nos atributos do sujeito, nos atributos do objeto, nas autorizações, nas obrigações e nas condições.

Os direitos podem ser divididos em: i) *Direito do Consumidor (CR)*: direito que um sujeito tem de consumir um determinado objeto. ii) *Direito do Provedor (PR)*: direito que um sujeito provedor tem de distribuir ou comercializar um objeto. iii) *Direito do Identificador (IR)*: direito que um sujeito identificador tem de acessar suas informações confidenciais.

As autorizações são processos que avaliam os atributos do sujeito, os atributos do objeto e os direitos solicitados juntamente com um conjunto de regras de autorização para determinar a decisão de uso. Posteriormente, retornam se o sujeito pode ou não exercer os direitos solicitados sobre um determinado objeto.

Uma autorização pode ser executada antes que um direito solicitado seja exercido, denominando-se pré-autorizações (*preA*). Outra forma de autorização é a contínua (*onA*), executada enquanto o direito é exercido. Em geral, a maioria das políticas de controle de acesso como MAC, DAC, RBAC e Gerenciamento de Confiança utilizam formas diferentes de pré-autorização. Certas autorizações podem necessitar fazer atualizações nos atributos de sujeitos ou objetos. Estas atualizações podem ser antes, durante ou depois da execução do direito de acesso.

As Obrigações são funções que verificam as exigências obrigatórias que um sujeito deve executar antes ou enquanto fizer uso de seus direitos. Uma obrigação pode utilizar alguns tipos de funções de histórico para verificar se certas atividades foram cumpridas adequadamente e então permitem ou não o acesso a um determinado objeto. Este tipo de

obrigação é chamado de pré-Obrigação (*preB*). Outro tipo de obrigação denomina-se Obrigação Contínua (*onB*). Trata-se de um predicado que deve ser satisfeito continuamente ou periodicamente enquanto os direitos de um sujeito estão em uso.

Os atributos de sujeito ou objetos podem ser usados para determinar que tipo de obrigações um determinado sujeito deve executar para que possa ter acesso a um objeto. No entanto, é importante salientar que os atributos não são utilizados pelas obrigações no processo de tomada de decisão, mas sim são utilizados apenas na escolha de que obrigações devem ser aplicadas. As obrigações podem causar certas alterações nestes atributos, os quais podem ser levados em consideração nas autorizações para as decisões de acesso atuais ou futuras.

As Condições são fatores de decisão que se baseiam no estado do sistema ou do ambiente, verificando se exigências relevantes são satisfeitas ou não. Assim, se as condições estiverem de acordo com a política de segurança, o acesso é permitido. Os atributos de sujeitos e objetos podem ser utilizados para selecionar qual exigência de condição deve ser utilizada para uma determinada solicitação de acesso. No entanto, não podem ser utilizados como exigências para o acesso. Elas não podem fazer atualizações em quaisquer atributos, sejam eles de sujeitos ou objetos. As Condições podem ser utilizadas na verificação do horário local para negar acesso em horários indevidos, ou então verificar os *status* do sistema e negar o acesso a todos os usuários caso seja verificado que o sistema esteja sobre ataque.

As Condições se diferenciam das Autorizações por avaliar as restrições do ambiente ou do sistema, características estas que não possuem relação alguma com os atributos de sujeitos ou objetos.

Baseando-se nos oito componentes do  $UCON_{ABC}$ , (PARK & SANDHU, 2004) desenvolveram um *framework* para classificar os modelos ABC. Estes modelos baseiam-se em três critérios distintos. O primeiro critério são os *fatores de decisão*, constituídos por



autorizações, obrigações e condições. O segundo é a *continuidade* no processo de decisão e, por fim, o terceiro é a *mutabilidade*, que permite alterações nos atributos de sujeitos e objetos diferentes vezes. Há um padrão numérico para a classificação de acordo com a mutabilidade dos atributos. Caso todos os atributos sejam imutáveis, o modelo adota o número identificador ‘0’. Para os atributos mutáveis, as pré-atualizações são identificadas pelo número ‘1’, as durante (*ongoing*) por ‘2’ e as pós-atualizações pelo número ‘3’. Os itens a seguir descrevem estas classificações. Baseando-se nesta numeração, estabeleceu-se uma tabela, representada pela Tabela 3.1, que possibilita a visualização dos modelos propostos.

	0 (imutável)	1 (pré-atualização)	2 (atualização-durante)	3 (pós-atualização)
<b>preA</b>	S	S	N	S
<b>onA</b>	S	S	S	S
<b>preB</b>	S	S	N	S
<b>onB</b>	S	S	S	S
<b>preC</b>	S	N	N	N
<b>onC</b>	S	N	N	N

Os casos não convenientes na prática são marcados com Não (N) e os que podem ser aplicados são marcados com Sim (S). Por exemplo, na primeira linha da tabela na Figura 3.4, é possível observar que, caso o fator de decisão seja feito antes do acesso (pré), as atualizações nos atributos podem ocorrer antes ou depois do usuário ter exercido seu acesso. Nas duas linhas mais inferiores, os atributos de condições são os únicos fatores de decisão. Por definição, os valores dos atributos das condições não podem ser atualizados, fazendo com que existam apenas dois tipos de modelos baseados em Condição. Considerando apenas os casos marcados com ‘S’, a Tabela 3.1 define os 16 modelos  $UCON_{ABC}$  básicos (PARK & SANDHU, 2004).

No controle de acesso tradicional, normalmente são focados os processos nos quais a decisão de uso é feita antes do direito solicitado ser exercido. Da mesma forma que nos modelos de controle de acesso tradicionais, nos modelos  $UCON_{preA}$ , as pré-Autorizações são utilizadas nos processos de decisão de uso.

Os modelos  $UCON_{preA}$  diferenciam-se de acordo com a mutabilidade de atributos, de forma que o  $UCON_{preA0}$  é um modelo de pré-autorização com atributos imutáveis que não necessitam de atualização. Este modelo possui exatamente os mesmos princípios que os modelos de Controle de Acesso Tradicionais como: MAC, DAC e RBAC. O  $UCON_{preA1}$  é um modelo de pré-autorização onde é possível que se façam atualizações em atributos antes que o sujeito exerça seus direitos. Por fim, ainda há o modelo de pós-atualização  $UCON_{preA3}$ , o qual atributos podem ser atualizados após um sujeito ter exercido seus direitos.

Um exemplo de pré-autorização é o sistema DRM *pay-per-view* com créditos pré-pagos. Neste tipo de sistema, os usuários pagam antecipadamente para ter créditos o suficiente para poder assistir determinada programação oferecida por um provedor de conteúdo. Neste caso, o modelo utilizado é o  $UCON_{preA1}$ . Assim, se o crédito que um sujeito possui é superior ao valor necessário para assistir uma determinada programação, o acesso à mesma é permitido. Uma vez que este acesso é permitido, o crédito do sujeito é reduzido de acordo com o valor de uso.

Devido à grande quantidade de modelos definidos na Tabela 3.1, é descrito apenas um exemplo para cada grupo de modelos.

Nos modelos  $UCON_{onA}$ , as solicitações de uso são permitidas sem qualquer tipo de pré-Autorização. As decisões de autorização são feitas de forma contínua, baseadas em tempo ou eventos, enquanto os sujeitos exercem seus direitos sobre os objetos. Em todas as vezes que as decisões de autorização forem feitas, o resultado deve ser positivo. A partir do momento em que a execução dos direitos de um determinado sujeito não satisfizer mais as políticas de controle de acesso do sistema, estes direitos são revogados e o acesso é negado.

Os modelos  $UCON_{onA}$ , são úteis para o uso de direitos relativamente longos. Segundo (PARK & SANDHU, 2004), todas as formas de mutabilidade de atributos são possíveis nos modelos  $UCON_{onA}$ . Assim, estes modelos compreendem: i)  $UCON_{onA0}$ : trata-se de um modelo com autorização contínua e sem procedimentos de atualização de atributos. Uma vez que não exista pré-Autorização, a solicitação de um acesso sempre é permitida. ii)  $UCON_{onA1}$ : é um modelo com autorização contínua e com procedimentos de atualização antes da execução dos direitos. iii)  $UCON_{onA2}$ : modelo com autorização contínua e com procedimentos de atualização de atributos durante a execução de direitos. iv)  $UCON_{onA3}$ : modelo com autorização contínua e com procedimentos de atualização de atributos depois da execução de direitos (PARK & SANDHU, 2004).

Um exemplo de autorização durante a execução é a limitação do número de uso simultâneo a um determinado objeto. Neste caso, utilizam-se os modelos  $UCON_{onA1}$ , e  $UCON_{onA3}$  para alterar os atributos do objeto. Em um sistema que possua um objeto que, por exemplo, apenas 10 sujeitos podem acessá-lo ao mesmo tempo, se o décimo primeiro sujeito solicitar acesso a este objeto, o primeiro sujeito que obteve acesso terá seu direito automaticamente negado. Assim, o décimo primeiro sujeito terá a permissão de acesso sem que haja o processo de pré-autorização. Para que este controle seja feito, o sistema deve monitorar o número de sujeitos que estão utilizando o objeto simultaneamente, assim como o tempo de início da solicitação de acesso de cada um destes sujeitos. Estas informações são os atributos dos objetos ( $ATT(O)$ ). O tempo de início de cada solicitação deve ser atribuído ao objeto antes que o sujeito inicie o direito de acesso. Após o sujeito terminar o acesso a um objeto, o sistema deve remover o atributo com a informação de tempo correspondente, como também diminuir o número de usuários que estão acessando este objeto.

Os modelos  $UCON_{preB}$  nada mais são do que obrigações que um sujeito deve cumprir antes que seja permitido o direito de acesso. Nestes modelos, uma solicitação pode exigir uma ou mais obrigações a serem cumpridas. O processo de pré-Obrigações ocorre em dois passos distintos. O primeiro seleciona quais são as obrigações a serem exigidas para uma determinada solicitação de uso. Tal seleção pode ser feita de acordo com os atributos de

sujeitos e objetos. O segundo passo é avaliar se as obrigações selecionadas foram cumpridas corretamente, para então permitir o acesso. As pré-Obrigações consistem em três modelos: i)  $UCON_{preB0}$ : modelo de obrigação que deve ser cumprido antes da permissão do acesso e que não atualiza atributos de sujeitos, nem de objetos. ii)  $UCON_{preB1}$ : este modelo é idêntico ao  $UCON_{preB0}$  exceto pelo fato de permitir atualizações em atributos de sujeitos e objetos antes da permissão do acesso. iii)  $UCON_{preB3}$ : é idêntico ao modelo  $UCON_{preB0}$  exceto pelo fato de permitir atualizações em atributos de sujeitos e objetos após a permissão do acesso (PARK & SANDHU, 2004).

Um exemplo de pré-obrigação, que faz uso do modelo  $UCON_{preB1}$ , é o caso no qual uma licença de uso deve ser aceita por um sujeito na primeira vez em que o mesmo acesse um objeto ou sistema. Neste caso, a licença de uso deve ser aceita apenas uma única vez. Para tanto, faz-se necessário a utilização de um atributo do sujeito, denominado “acordo”. Uma vez que o sujeito tenha concordado com a licença de uso na primeira vez em que o mesmo for entrar no sistema, este atributo é configurado como *true*. Caso a licença não tenha sido aceita, este atributo é configurado como *false* e o sujeito não terá permissão para entrar no sistema. Todas as vezes que ele tentar entrar no sistema, novamente a licença de uso lhe aparecerá como uma obrigação para que ele possa ter acesso ao sistema, até que ele a aceite.

Os modelos  $UCON_{onB}$  são similares aos modelos  $UCON_{preB}$ , exceto pelo fato de que as obrigações devem ser cumpridas periodicamente ou continuamente enquanto os direitos são exercidos. Para que isto seja possível, faz-se necessário a utilização de um parâmetro de tempo  $T$  como parte dos elementos de obrigação. O parâmetro  $T$  define os intervalos de tempo que podem ser baseados em tempo ou eventos. Existem quatro tipos de modelos de obrigação durante a execução. O primeiro deles é o  $UCON_{onB0}$ , que não permite a atualização de atributos. Os modelos  $UCON_{onB1}$ ,  $UCON_{onB2}$  e  $UCON_{onB3}$  são similares ao modelo  $UCON_{onB0}$ , com a diferença de que eles possibilitam pré - atualizações, atualizações durante a execução e pós- atualizações, respectivamente.

Um exemplo da aplicação do  $UCON_{onB}$  pode ser um servidor de Internet livre que exige que seus usuários vejam anúncios enquanto os mesmos estão conectados como servidor. Contanto que a janela de anúncios esteja ativa, o uso da conexão é permitido. A frequência com que a verificação se a janela está ativa ou não é feita, ou então a forma como esta verificação é feita (baseada em intervalo de tempo ou eventos), fica de acordo com o desejo ou necessidade do servidor.

O modelo de pré-Condição inclui certas restrições de ambiente que não se relacionam diretamente com sujeitos ou objetos. As pré-condições devem ser satisfeitas antes que os direitos de acesso sejam exercidos. Todas as vezes que as condições do ambiente são avaliadas para que se possa ou não permitir um acesso, faz-se necessária uma busca pelo *status* atual do ambiente. Diferentemente dos modelos de autorizações e obrigações, os atributos de sujeitos e objetos no  $UCON_{preC}$  não podem ser alterados. Isto se deve ao fato de que o valor do *status* pode ser alterado apenas de acordo com a situação atual do ambiente, como por exemplo, a hora do dia, a localização na rede através de endereço IP ou a quantidade de usuários no sistema. Assim, o  $UCON_{preC}$  possui apenas um modelo:  $UCON_{preCo}$ . Apesar dos atributos de sujeitos ou objetos não serem usados no processo de decisão de acesso, eles podem ser utilizados para decidir que tipos de elementos de condição devem ser utilizados para estabelecer a decisão de acesso (PARK & SANDHU, 2004).

Um exemplo de pré-condição é a restrição de locais em que determinados direitos podem ser exercidos. Isto pode ser feito através da verificação de um identificador de CPU ou então um endereço IP antes de uma permissão de acesso. Assim, é possível permitir que apenas IPs provenientes de universidades conveniadas tenham acesso a conteúdos de bibliotecas digitais.

O modelo  $UCON_{onC}$  tem por característica o monitoramento contínuo das condições de ambiente enquanto os direitos de acesso estão em uso. Caso tais condições não satisfaçam

as restrições estabelecidas pelo sistema, os direitos de acesso do sujeito são negados. Neste modelo, o acesso a um objeto é permitido sem qualquer processo de decisão no momento da solicitação. Da mesma forma que no modelo  $UCON_{preC}$ , o  $UCON_{onC}$  também não permite mutabilidade nos atributos de sujeitos e objetos, possibilitando assim a existência de apenas um modelo: (PARK & SANDHU, 2004).

Um exemplo da aplicação do modelo pode ser a limitação de tempo permitido acesso de um sujeito a um objeto. Supondo que em um sistema um sujeito possa acessar um objeto apenas em um determinado período do dia (entre às 08:00 e às 18:00 horas). O horário atual do dia é um *status* do ambiente local, e não um atributo do sujeito ou objeto. Para que o controle de acesso seja feito sobre este sujeito, o monitoramento do horário de acesso permitido pelo sujeito de acessar um objeto é monitorado continuamente fazendo sempre a comparação com o horário atual. Caso o horário atual se encontre fora dos limites pré-estabelecidos, os direitos de uso do sujeito são automaticamente negados.

A definição de uma arquitetura para o UCON é uma tarefa muito complexa pelo fato de existirem vários fatores de variação. Em (PARK, 2003), foram levados em consideração dois fatores: a existência de funções de pagamento e a localização do monitor de referência. Através destes fatores, Park definiu oito arquiteturas que não são exibidas nesta dissertação pelo fato de possuírem características relacionadas a sistemas que nada tem a ver com o de CE B2B.

Nas seções a seguir são descritas as arquiteturas que influenciaram (PARK, 2003) a definir suas oito arquiteturas.

Segundo (PARK, 2003), o controle de acesso a recursos digitais, do ponto de vista de pagamentos, pode ser dividido em dois tipos: Tipo Baseado em Pagamento (PBT – *Payment Based Type*) e Tipo Livre de Pagamento (PFT – *Payment Free Type*).

A arquitetura PBT tem por objetivo distribuir o maior número possível de informações digitais para, posteriormente, solicitar o pagamento para cada uma das cópias. Neste tipo de arquitetura faz-se necessário a utilização de uma função de pagamento para que se possa acessar uma informação digital distribuída. Em muitos casos é aceitável, ou até mesmo desejado, que uma pequena quantidade de informações sejam acessadas irregularmente para fins de propaganda do conteúdo.

Um dos maiores problemas da arquitetura PBT são as brechas de segurança na compra dos direitos de acesso de informações digitais, que têm por consequência direta muitos prejuízos financeiros. Devido a fato da redistribuição de informações digitais adquiridas ilegalmente não reduzir sua qualidade ou valor, provedores de conteúdos têm se esforçado para desenvolver tecnologias de combate à distribuição não-autorizada destas informações.

Neste tipo de arquitetura, a distribuição de informação digital não exige pagamento. No entanto, esta distribuição deve ser controlada no intuito de prover a confidencialidade ou outras exigências de segurança para estas informações. Ela não permite qualquer tipo de perda ou roubo de informações entre a origem e o destino.

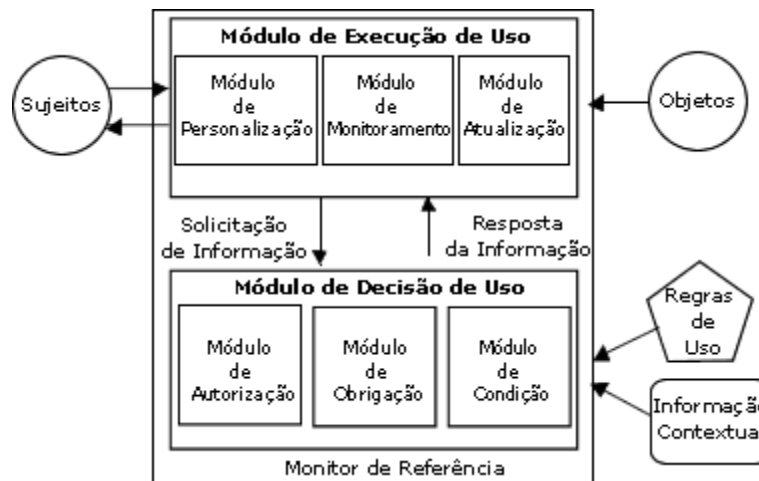
A arquitetura PFT é utilizada em situações onde a segurança da informação está em primeiro lugar. Um exemplo pode ser o envio de documentos ultra-secretos entre governos de dois países aliados de forma que estas informações não sejam reveladas, intencionalmente ou acidentalmente, para outros países inimigos.

(Park, 2003) faz a observação da importância de haverem estudos na definição de novas arquiteturas de segurança generalizadas que possam prover ambientes seguros em PFT, assim como prover a base para o desenvolvimento de soluções de controle de acesso que satisfaçam as exigências das organizações.

Um Monitor de Referência é um conceito básico que fornece mecanismos de controle de acesso a informações digitais através da associação entre políticas de decisão e regras.

Assim, Sujeitos podem acessar Objetos digitais apenas através do intermédio de um Monitor de Referência. Ele tem sido muito discutido pela comunidade de controle de acesso por ser um dos assuntos mais críticos, do ponto de vista arquitetural.

Segundo (SANDHU & PARK, 2003), o Monitor de Referência UCON sem diferencia em alguns detalhes do padrão proposto pela ISO [ISO/IEC 10181-3]. A Figura 3.4 ilustra a estrutura conceitual deste Monitor de Referência que é constituído de dois módulos:



- possui os módulos de autorização, obrigação e condição. O Módulo de Autorização é responsável por um processo similar aos processos de autorização tradicional que utilizam os atributos de sujeitos e objetos juntamente com as regras de uso para verificar se uma solicitação é permitida. Ele pode retornar como resposta um ‘sim’ ou ‘não’, ou então meta-dados com informações dos objetos que podem ser acessados juntamente com os direitos permitidos. O Módulo de Condição tem por finalidade decidir, para as solicitações autorizadas, se as exigências condicionais estão de acordo ou não com as regras de uso. O Módulo de Obrigação decide se certas obrigações devem ser executadas ou não, antes ou durante o acesso.
- possui os módulos de personalização, monitoramento e atualização. O Módulo de Personalização é utilizado quando as informações dos meta-dados, provenientes do Módulo de Autorização, são utilizadas



para a personalização do objeto solicitado. O Módulo de Monitoramento tem por objetivo monitorar constantemente o Módulo de Obrigação para ver se existe alguma obrigação que deva ser cumprida. O Módulo de Atualização é sempre acionado a partir do resultado do Módulo de Monitoramento.

Baseando-se na localização do Monitor de Referência, (SANDHU & PARK, 2003) definiram dois tipos básicos de arquitetura: Monitor de Referência do lado do Servidor (SRM – *Server-side Reference Monitor*) e Monitor de Referência do lado do Cliente (CRM – *Client-side Reference Monitor*). Nesta definição, é levado em consideração que um Servidor nada mais é do que uma entidade que provê um processo ou objeto digital, e Cliente uma entidade que executa um processo ou acessa um objeto digital.

Esta é uma arquitetura tradicional onde o Monitor de Referência está localizado no lado do Servidor e intermedia todos os acessos a objetos digitais. Ela é normalmente utilizada por modelos de controle de acesso tradicionais e Gerenciamento de Confiança. Um sistema com este tipo de arquitetura facilita o controle de acesso de sujeitos a objetos digitais devido a existência de um controle central (SANDHU & PARK, 2003).

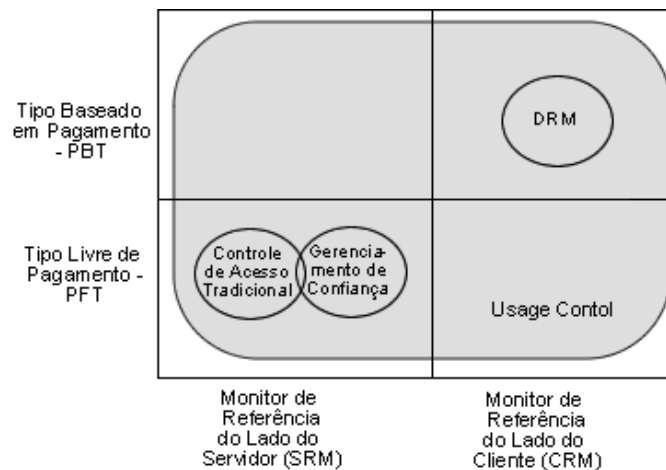
Neste tipo de ambiente, objetos digitais podem ser ou não armazenados no lado do cliente. Caso sejam, isto significa que não é objetivo do UCON controlar a disseminação destes objetos, podendo ser utilizados e modificados livremente no lado do cliente. No entanto, caso não seja permitido que estes objetos sejam armazenados no lado do cliente, isto significa que eles devem protegidos e controlados centralmente, ou seja, sempre serem armazenados no lado do servidor.

A arquitetura CRM vem sendo utilizada em soluções DRM. Nela, o Monitor de Referência localiza-se no lado do cliente no intuito de controlar o acesso a informações digitais já distribuídas. Ela possui um nível de confiança mais baixo que a arquitetura SRM. Devido a isto, ela é mais apropriada a aplicações com exigências de garantia menores.

O uso de objetos digitais armazenados no lado do cliente na arquitetura CRM faz com que eles estejam protegidos pelo Monitor de Referência local ao invés do servidor. Desta forma, um sistema CRM é apropriado a sistemas de Comércio Eletrônico B2C com grande quantidade de distribuição de informações digitais como arquivos de MP3 ou *e-books*.

Esta arquitetura mista é capaz de fornecer um controle de acesso em duas camadas. Na primeira, a arquitetura SRM pode ser utilizada para a distribuição de objetos digitais com um controle relativo. Na segunda camada, a arquitetura CRM pode ser utilizada para um controle de acesso minucioso. Para implementações do mundo real, as arquiteturas SRM e CRM são implementadas em conjunto para de prover maior funcionalidade e segurança.

A Figura 3.5 ilustra a abrangência do UCON em relação a outros modelos de controle de acesso quando se é levado em consideração às arquiteturas mencionadas nesta seção.



Este Capítulo apresentou o modelo UCON, um novo e abrangente modelo de controle de acesso proposto, de forma completa, por Jaehong Park e Ravi Sandhu em meados de 2003. Sua principal contribuição para a comunidade científica é unificação de diversas

áreas de controle de acesso, além de introduzir novos conceitos como obrigação, condição, continuidade e mutabilidade.

Park e Sandhu afirmam existem ainda muitas melhorias a serem pesquisadas e desenvolvidas. Dentre elas, está o estudo em segurança para sistemas B2B. Desta forma, o próximo Capítulo falará sobre sistemas de CE B2B para que se tenha uma idéia conceitual deste tipo de comércio eletrônico e, conseqüentemente, possamos estabelecer modelos de implementação para este tipo de sistema, baseando-se no  $UCON_{ABC}$ .

Este capítulo fornece uma visão geral sobre sistemas de Comércio Eletrônico *Business-to-Business* (B2B), abordando conceitos fundamentais e aspectos de interação.

(BLODGET & MCCABE, 2000) afirmam que o termo “*business-to-business*” teve início na década de 60. Naquela época, parceiros comerciais realizavam troca de dados através de linhas telefônicas utilizando um formato de dados proprietário previamente estabelecido entre ambas as partes. Este método ajudou a melhorar a eficiência nas transações de negócios entre as empresas parceiras. No entanto, esta tecnologia possuía a desvantagem de os formatos dos dados das empresas parceiras variarem muito, dificultando assim o pré-estabelecimento dos dados entre as empresas parceiras.

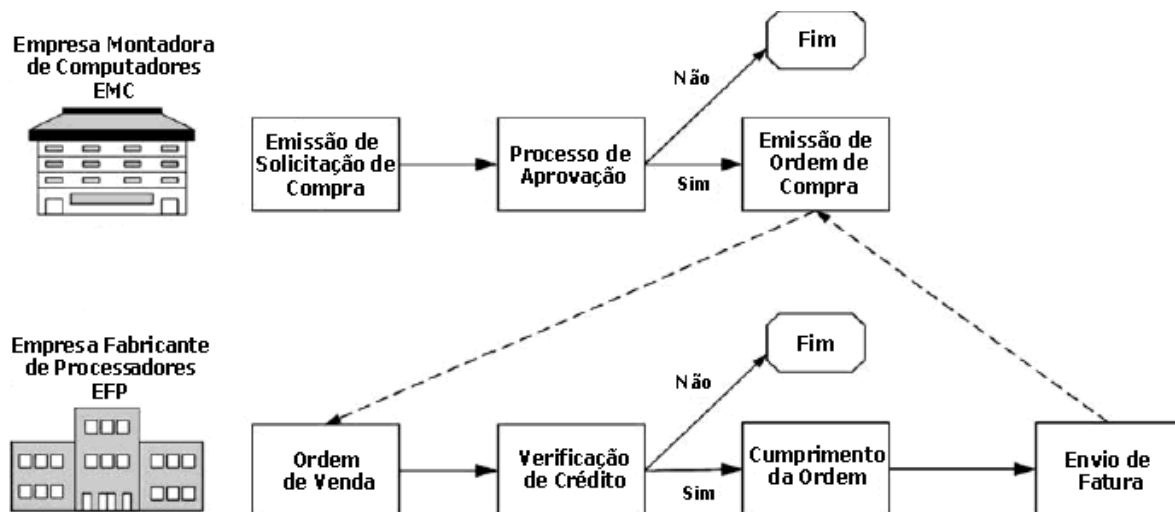
EDI (*Electronic Data Interchange*) surgiu na década de 70 como forma de transmissão de dados padronizados que agilizaram a execução dos processos entre as empresas parceiras. Esta tecnologia era utilizada sobre redes de comunicação privadas denominadas VANs (*Value Added Networks*). Este tipo de rede possui um custo de instalação e manutenção muito elevado, excluindo assim a possibilidade de sua utilização em empresas de pequeno e médio porte.

Na década de 90, o Comércio Eletrônico B2B teve um crescimento significativo devido surgimento e a popularização da Internet. A Internet possibilitou que empresas de pequeno e médio porte aderissem ao CE B2B. Além disso, também possibilitou que empresas de grande porte, que já possuíam sistemas baseados em EDI, reduzissem seus custos e expandissem o número de interações com outras empresas parceiras.

(BRODIE, 2000) define este tipo de CE como o uso de sistemas computadorizados para a administração de negócios entre diferentes empresas parceiras. Segundo (MEDJAHED et All, 2003), milhões de empresas já migraram ou estão migrando suas principais operações

para a Internet no intuito de obter as inúmeras vantagens que ela pode proporcionar. Com o CE B2B surgiram novas possibilidades de comercialização, permitindo que empresas de diferentes cidades, países ou até mesmo continentes interagissem e cooperassem entre si para realizarem transações de forma mais eficiente, encontrassem novos parceiros ou compartilhassem experiências.

A Figura 4.1 ilustra o funcionamento básico de um sistema B2B descrito por (MEDJAHED et All, 2003). Através desta figura, é possível observar a existência de duas empresas parceiras: i) A *Empresa Montadora de Computadores* (EMC), que monta e vende computadores; ii) A *Empresa Fabricante de Processadores* (EFP), que é responsável pela fabricação e fornecimento de todos os processadores inseridos nos computadores fabricados pela EMC. Estas empresas automatizam seus processos de compra e venda de produtos entre si através de um sistema de CE B2B. Neste sistema, um usuário da EMC envia uma ordem de compra para a EFP. Uma vez que esta ordem chegue no sistema de processamento de ordens da EFP, a ordem de compra é transformada em uma ordem de venda. Posteriormente é realizada a verificação de crédito onde, em caso positivo, um cumprimento de ordem é emitido pela EFP e uma fatura é enviada à EMC.



## 4.2. E-MARKETPLACES

Quando se fala em CE B2B, é comum imaginar sistemas que realizam processos empresariais entre duas empresas parceiras. No entanto, existe uma outra forma de CE B2B, denominado Mercado Eletrônico (*Electronic Marketplace* ou *E-Marketplace*). Segundo (QUIX, SCHOOP & JEUSFELD, 2002), este tipo de sistema fornece um fórum que viabiliza a reunião de várias empresas compradoras e vendedoras com o objetivo de possibilitar a troca de informações sobre produtos e processos e apoiar transações comerciais.

(SCHOOP, KOLLER & QUIX, 2001) definiram três fases distintas no desenvolvimento de *E-Marketplace*. A primeira fase é a *busca* por novos parceiros empresariais. Esta fase tem por objetivo encontrar informações relevantes sobre potenciais parceiros comerciais como, por exemplo, um comprador que deseja encontrar fornecedores para produtos que procura. Para atender tal objetivo, um *e-marketplace* deve oferecer ferramentas de busca sofisticadas que se baseiem em outros fatores além de palavras-chave, como por exemplo: suporte a diferentes ontologias, línguas ou terminologias. A segunda fase é a de *negociação* que conduz ao estabelecimento de um contrato. É nesta fase que parceiros negociam preços, definem datas de entrega ou negociam sobre a qualidade do produto negociado. Assim, caso haja um acordo entre ambas as partes, é estabelecido um contrato. Esta fase necessita de protocolos de negociação complexos devido à grande quantidade de trocas interativas de mensagens e documentos necessários para o estabelecimento da negociação. Por fim, a última fase é a de *cumprimento* dos termos estabelecidos pelo contrato. Um fornecedor pode ter motivos para o atraso na data de entrega de um determinado produto. Em caso deste fornecedor se sentir prejudicado por algum item do contrato e sugerir alterações, o sistema deve fornecer meios para que as duas partes discutam a alteração deste contrato e cheguem a um acordo. (QUIX, SCHOOP & JEUSFELD, 2002) afirmam que este tipo de suporte é uma característica inovadora que contribui em muito às praticas atuais de negociação entre as empresas.

Em um estudo recente, (RADOWIISKY,2002) afirma que a idéia de junção entre sistemas de CE B2B e sistemas de Gestão Empresarial (*Enterprise Resource Planning - ERP*) surgiu da grande necessidade de integrar completamente e automatizar todos os fluxos dos processos de uma empresa, tornando-os mais rápidos e eficientes. Esta integração pode responder perguntas como: “Eu tenho produtos o bastante no estoque para que eu possa entregá-los na hora certa?”. Esta junção proporciona um ambiente completo em que, além de ser possível realizar a gestão dos processos da própria empresa, há também a possibilidade de realizar transações comerciais com empresas parceiras. A interação possui um papel importante neste tipo de sistema, pois ela é uma “ponte” de ligação entre o sistema de uma empresa e o sistema de cada empresa parceira, independentemente da tecnologia utilizada. Diversos trabalhos publicados, como os de (MEDJAHED et All, 2003), (DABOUS, RABHI & RAY, 2002) ou (QUIX, SCHOOP & JEUSFELD, 2002), enfatizam o estudo em métodos ou tecnologias necessárias para a interação entre sistemas B2B, confirmando assim que a interação nestes tipos de sistemas já é uma realidade. (DABOUS, RABHI & RAY, 2002) afirmam que a integração é um desafio devido as seguintes fatores:

- O formato das informações está cada vez mais se diversificando;
- O espaço ocupado pelas informações é cada vez maior e dinâmico;
- A integração semântica dos dados é muito complexa;
- A maioria dos sistemas é autônoma;
- A integração precisa ser simples, rápida, segura e adaptável a mudanças.

O CE B2B compreende uma grande variedade de interações entre parceiros de negócio. Segundo (MEDJAHED et All, 2003), os tipos de interações dependem dos cenários de uso, das partes envolvidas e das exigências empresariais. Desta forma, é de fundamental importância determinar as exigências relevantes para a definição de um modelo de

interações. Este modelo é utilizado no desenvolvimento de um sistema B2B. Em sua pesquisa, Medjahed destacou os seguintes fatores:

- refere-se ao nível de relacionamento e duração da parceria entre duas empresas. Levando-se em consideração o nível de relacionamento, dois parceiros são *fortemente unidos* quando possuem forte dependência um do outro e *fracamente unidos* quando há troca de informações empresariais sob demanda. Levando-se em consideração a duração da parceria, o relacionamento entre parceiros é *dinâmico*, onde os negócios podem precisar formar uma sociedade rápida e curta, ou *longo*, em que o negócio define previamente uma sociedade duradoura.
- refere-se ao nível de diferença entre parceiros empresariais. Sistemas utilizados nas empresas utilizam diferentes tecnologias e tipos de estruturas de dados. Quanto maior a interconexão destes sistemas, maior a complexidade no desenvolvimento de sistemas de CE B2B.
- refere-se à possibilidade dos sistemas dos parceiros envolvidos serem autônomos em seu desenvolvimento, comunicação e execução. Assim, cada parceiro pode escolher a tecnologia a ser utilizada, os modelos de descrição de conteúdo, os modelos de programação ou os modelos de interação. Tal característica permite um melhor controle sobre a implementação de seus sistemas, podendo ter a flexibilidade de mudar seus processos sem afetar outros parceiros.
- refere-se ao nível de visibilidade externa e gerenciabilidade das aplicações de parceiros. Esta característica deve facilitar a supervisão e controle da execução, medição de performance e o prognóstico da disponibilidade e *status* dos sistemas B2B de cada parceiro.
- refere-se ao nível em que uma aplicação é capaz de se adaptar rapidamente a mudanças. Este fator possui muita importância devido ao ambiente altamente dinâmico que operam os sistemas B2B. Assim, novos processos *on-line* podem surgir e tornarem-se necessários, assim como outros podem tornar-se desnecessários e serem retirados do sistema. Em geral, o impacto das mudanças depende do nível de autonomia entre as aplicações.



- a segurança é uma das maiores preocupações em sistemas de CE. Medidas de segurança sofisticadas devem sempre ser colocadas em prática no intuito de assegurar aos parceiros empresariais que suas transações estão seguramente controladas. Para atingir um nível de segurança adequado, as aplicações B2B devem suportar: controle de acesso, autenticação, autorização, integridade na comunicação, confidencialidade e não-repúdio.
- refere-se à capacidade de crescer de um sistema em uma ou mais dimensões devido a fatores como: o aumento no volume de dados acessíveis, aumento do número de transações que podem ser feitas em uma determinada unidade de tempo ou o aumento no número de relacionamentos que podem ser suportados.

A interação em sistemas B2B é feita em três tipos diferentes de camada, as quais podem utilizar diferentes tipos de tecnologia. As camadas são: i) Camada de Transporte; ii) Camada de Conteúdo; iii) Camada de Processos Empresariais (DABOUS, RABHI & RAY, 2002).

Esta camada tem por preocupação a interação na troca de mensagens entre parceiros localizados remotamente e que utilizam diferentes protocolos de comunicação. Assim, esta camada visa proporcionar independência de protocolos através da tradução e conversão de mensagens entre protocolos heterogêneos através da utilização de *gateways*. Nesta camada a comunicação entre parceiros pode ser feita de duas formas: i) Através de Protocolos de Rede que são utilizados para transferir dados brutos; ii) Através de Chamadas de Procedimentos Remotos (*Remote Procedure Call - RPC*), utilizados na implementação de Componentes.

Dos *Protocolos de Rede*, o primeiro protocolo a ser utilizado em aplicações B2B surgiu juntamente com o primeiro e mais utilizado *framework* de interação de dados: EDI. Este *framework* trabalhava sobre redes de computadores dedicadas, as VANs. As VANs são

utilizadas para controlar a entrega de mensagens e direcioná-las entre parceiros empresariais.

Os Protocolos TCP/IP são os protocolos padrão na Internet. Eles permitem a comunicação entre vários sistemas através de diferentes sistemas operacionais, plataformas e linguagens de programação. Desta forma, eles se tornaram um padrão, entre os protocolos de rede, na transferência de dados reduzindo em muito o custo para o desenvolvimento de sistemas B2B.

Os *Componentes* se tornaram as principais metodologias no desenvolvimento de sistemas distribuídos. O desenvolvimento de sistemas de CE B2B baseado em componentes é mais apropriado quando há um pequeno número de parceiros. Os componentes abrangem principalmente as comunicações em nível de transporte, pois exibem capacidades limitadas de interações na camada de conteúdo. Os componentes são módulos de programas que podem ser desenvolvidos e distribuídos independentemente. Eles são como caixas-pretas que fornecem acessos aos seus processos através da definição de uma interface. Assim, eles têm sido utilizados na computação distribuída de forma que um objeto (componente) pode acessar outro objeto remoto. A interação entre estes objetos só é possível através de um componente *middleware*, que é a estrutura que dá suporte à criação, desenvolvimento e interação entre os componentes (DABOUS, RABHI & RAY, 2002). Segundo (MEDJAHED et All, 2003), os maiores componentes *middleware* atualmente são: i) Componentes baseados em Java como *Java Remote Method Invocation* (RMI) ou *Enterprise Java Bean* (EJB). ii) *Common Object Request Broker Architecture* – CORBA. iii) *Distributed Component Object Model* – DCOM.

Esta camada provê linguagens e modelos para descrever e organizar informações, de diferentes tecnologias, de forma que elas possam ser entendidas e utilizadas. Sua preocupação está em resolver problemas relacionados a assuntos de heterogeneidade das informações em nível semântico, nos quais há diferentes interpretações de um mesmo conceito, e estrutural, onde há o uso de diversos formatos de informações. Desta forma, ela pode fornecer independência de modelos de dados, formatos e linguagens. Para que seja

possível a integração, tanto quem envia quanto quem recebe informações, deve concordar inicialmente no formato compartilhado. Normalmente, o sistema de quem envia uma mensagem possui métodos que codificam o seu conteúdo em um formato previamente definido, para então poder enviá-la. O sistema receptor interpreta a mensagem para recuperar o formato adequado ao sistema. Desta forma, a definição de um formato padrão pode vir a permitir que novos sistemas empresariais não se preocupem com a interpretação na troca de dados (DABOUS, RABHI & RAY, 2002).

A linguagem XML (*eXtensible Markup Language*) é uma tecnologia de padrões abertos utilizada na definição, armazenamento e recuperação de dados estruturados. Devido ao seu grande potencial para ser no futuro um formato padrão para a transferência de dados e comunicação entre objetos em sistemas distribuídos, um grande número de *frameworks* de interação em sistemas B2B é baseado em XML. Desta forma, uma empresa pode criar e publicar documentos XML que descrevam suas ofertas, exigências, suposições e termos para que se possa realizar negócios. Os parceiros em um sistema B2B devem então interagir entre si apenas depois de inspecionar e entender as descrições dos documentos XML de cada um. (MEDJAHED et All, 2003) afirma que o objetivo desta técnica é permitir o uso de processos na *Web* sem que haja custos na modificação de padrões entre os sistemas parceiros. Existem vários *frameworks* para interação entre sistemas B2B baseados em XML. Pesquisas como as de (DOGAC & CINGIL, 2002) ou (MEDJAHED et All, 2003) fazem um estudo comparativo destes *frameworks*, dos quais os principais citados são: eCO BizTalk cXML RosettaNet e ebXML.

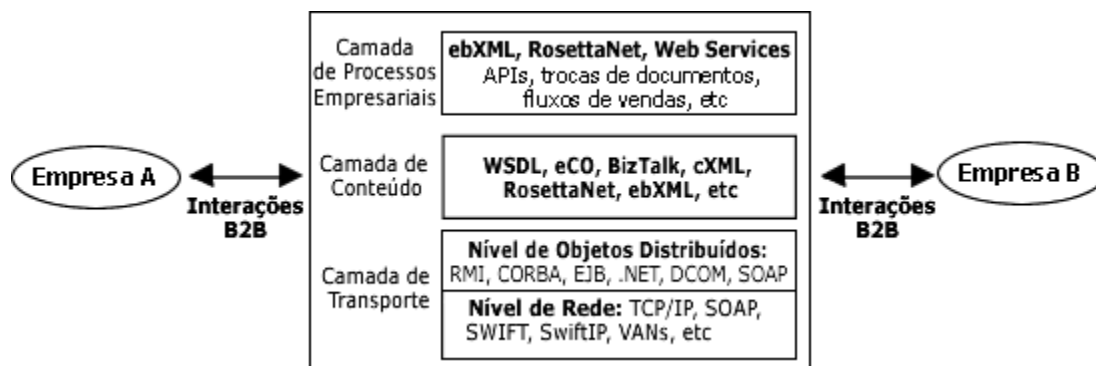
Estes *frameworks* fornecem suporte para a integração na Camada de Conteúdo. Entretanto, a maioria deles, como ebXML, RosettaNet ou eCO, também oferecem suporte para a integração na Camada de Processos Empresariais.

(MEDJAHED et All, 2003) afirma que a Camada de Processos Empresariais se preocupa com a semântica de interações entre os processos empresariais em comum entre duas empresas parceiras. O objetivo destas interações é que parceiros autônomos e heterogêneos façam uso da *Web* para anunciar seus termos e capacidades e estabelecer uma

relação com outros parceiros através da interação de seus processos empresariais. Desta forma, se torna possível estabelecer interações ponto-a-ponto, como por exemplo: troca de contratos, realização de pedidos de mercadorias, dentre outros. As interações nesta camada devem ser transparentes, de forma que seus processos não se preocupem com as tecnologias utilizadas no conteúdo das informações, ou então com a forma em que estas informações irão trafegar na rede. A interoperabilidade nesta camada é um desafio muito difícil de ser alcançado devido à necessidade de se entender a semântica do processo empresarial de cada parceiro. Segundo (DABOUS, RABHI & RAY, 2002), potenciais soluções para este desafio são: i) Interface de Programação de Aplicação (*Application Programming Interface* - API); ii) Soluções baseadas em documento; iii) Soluções Baseadas em *Workflow* e iv) *Web Services*.

As *APIs* tem por objetivo determinar as conexões globais, coordenar as operações empresariais e definir as interfaces abstratas que possibilitem invocações de operações remotas. Tecnologias de Banco de Dados e *middleware* são utilizadas para o mapeamento destas interfaces abstratas para implementações físicas. Já nas *soluções baseadas em documento*, um conjunto de documentos é trocado de acordo com um protocolo, de forma que não exista acordo prévio entre os parceiros. Desta forma, torna-se possível que cada parceiro publique seus documentos de forma independente. Cada documento publicado é autodescritivo e contém informações suficientes sobre o processo empresarial envolvido. As *soluções baseadas em workflow* são ineficientes quando são consideradas as necessidades do CE B2B como, por exemplo, o relacionamento complexo entre parceiros. Desta forma, o objetivo dos sistemas *workflow* entre empresas é automatizar os processos empresariais que interconectam e gerenciam as conexões entre sistemas parceiros. Este tipo de sistema é um conjunto de atividades que são implementadas por diferentes empresas e que representam um processo empresarial que ultrapassa as fronteiras das empresas envolvidas. Por fim, os *Web Services* são ferramentas de interação que vêm crescendo continuamente em aplicações baseadas na internet. Com *Web Services* a interação é feita através da troca de mensagens baseadas em documentos XML, possibilitando que aplicações Web sejam subdivididas em pequenos processos compartilhados (DABOUS, RABHI & RAY, 2002).

Após a descrição das três camadas de interação em sistemas de CE B2B, é possível compreender a localização das tecnologias de interação existentes atualmente através da Figura 4.2. (MEDJAHED et All, 2003) descreve cada uma destas tecnologias.



Os sistemas de CE B2B vêm sendo utilizados há algum tempo por empresas para agilizar seus processos juntamente com seus parceiros comerciais. A junção do CE B2B com sistemas ERP é uma tendência promissora e que tende a se desenvolver cada vez mais. Após realizar uma pesquisa nesta área, foi possível perceber que ainda hoje existem dois grandes desafios em aberto. O primeiro deles é a integração de processos e informações entre sistemas de diferentes tecnologias, plataformas, linguagens ou línguas. O segundo desafio, e mais importante para esta dissertação, é a questão da segurança. A segurança deve ser obrigatória para dar às empresas a certeza de que suas transações são controladas de forma segura. (MEDJAHED et All, 2003) afirmam que são necessárias pesquisas na especificação, validação e execução de políticas de controle de acesso para sistemas B2B. De acordo com estas necessidades, no capítulo 5 é apresentada uma pesquisa na área de Controle de Acesso em sistemas de CE B2B. Tal pesquisa tem o intuito de relatar os esforços que vem sendo feito atualmente para que se desenvolvam modelos, *frameworks*, ou esquemas mais adequados às necessidades do CE B2B.

Existem diversos esforços no sentido de definir teorias, técnicas, modelos ou *frameworks* mais adequados para a implementação do controle de acesso e segurança em sistemas de CE B2B. Este capítulo faz referência a alguns destes trabalhos. Desta forma, será possível compreender o que vem sendo feito nesta linha de pesquisa para então contextualizar melhorias utilizando o modelo UCON<sub>ABC</sub>.

(GOODWIN, GOH & WU, 2002) afirmam que sistemas de controle de acesso modernos, voltados para sistemas de comércio eletrônico, devem ser simples, concisos, com definições de políticas de controle de acesso facilmente configuráveis que estejam alinhadas com os processos empresariais. Desta forma, eles descrevem um esquema para sistemas de *E-marketplace*, denominado Controle de Acesso Baseado em Política, que se baseia no modelo RBAC. Atualmente, este esquema é implementado como um mecanismo de controle de acesso, em nível de aplicação, no produto da IBM denominado: *WebSphere Commerce Suite Marketplace Edition*.

Segundo (GOODWIN, GOH & WU, 2002), um *E-marketplace* define um conjunto de processos e processos empresariais, cadastra empresas participantes e lhes dá acesso para um determinado subconjunto destes processos e processos. Assim, as políticas de controle de acesso do responsável pelo *E-marketplace* define quais ações estas empresas podem executar. Cada empresa participante pode atribuir permissões de acessos a seus empregados baseando-se em suas próprias políticas de segurança.

O primeiro objetivo de (GOODWIN, GOH & WU, 2002), no desenvolvimento deste esquema, foi melhorar o modelo RBAC para que ele atendesse às necessidades das aplicações de CE B2B. Eles afirmam que o modelo RBAC possui a limitação de que a atribuição de papéis a sujeitos é a única forma de atribuir-lhes permissões. Esta técnica torna-se inadequada para alguns casos de sistemas de CE B2B. Um exemplo disto é o caso

onde, em um sistema B2B entre duas empresas (Alfa e Beta), existam funcionários administradores de contratos que devem possuir permissões diferentes em cada empresa. Neste caso, torna-se necessária a criação de dois papéis distintos: AdmContratoAlfa e AdmContratoBeta. Além disso, se a empresa Alfa for uma multinacional e as leis dos países que atua forem diferentes, isto exigiria que os administradores de contratos de países diferentes tivessem permissões diferentes. Desta forma, seria necessário criar outros papéis como: AdmContratoAlfaUSA e AdmContratoAlfaCanada. Com isso, os nomes dos papéis determinariam os sujeitos que um administrador deveria atribuir estes papéis. Assim, estes papéis deixariam de ser genéricos para se tornarem papéis específicos que poderiam ser atribuídos a um número reduzido de sujeitos.

Para resolver este problema (GOODWIN, GOH & WU, 2002) utilizam o agrupamento implícito de sujeitos e utilizam estes grupos para mapear os sujeitos a permissões em grupos de objetos. Um grupo implícito é definido por um conjunto de restrições e, qualquer sujeito que satisfaça estas restrições, é um membro do grupo. O papel AdmContratoAlfaUSA, por exemplo, é equivalente a um grupo com as restrições: [(empresa = Alfa), (país = USA) e (cargo = Administrador de Contrato)]. As restrições deste exemplo nada mais são do que atributos do sujeito. Estas restrições permitem que membros de grupos sejam determinados eficientemente através dos atributos dos sujeitos e da definição do grupo. Se um funcionário de uma empresa com o cargo de administrador de contratos, por exemplo, é transferido para outro país na empresa em que trabalha, então a atualização do ambiente de trabalho é feita automaticamente fazendo com que este funcionário mude de grupo sem interferência de um administrador. Caso a definição de um grupo se torne dependente de outra restrição, basta definir o novo grupo e atribuí-lo às permissões necessárias. O agrupamento implícito também pode ser aplicado a objetos, os quais podem ser agrupados por tipo ou estado.

No desenvolvimento de objetos de negócio é importante manter relacionamentos ou associações entre objetos ou entre objetos e usuários. Estes relacionamentos podem ser de cardinalidade de um-para-muitos ou muitos-para-muitos. Existem métodos que podem ser

utilizados para determinar o relacionamento entre sujeitos e objetos, ou então determinar se um determinado sujeito é o criador do objeto.

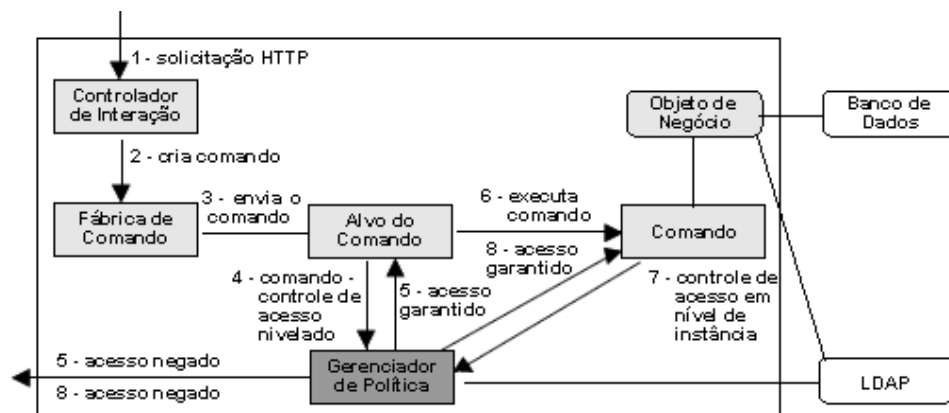
Para cada recurso que pode ser controlado no sistema é necessário que haja um proprietário para determinar qual política de controle de acesso deve ser aplicada a um determinado objeto. As políticas definidas podem determinar quais sujeitos podem ter acesso a quais objetos. Um proprietário pode ser tanto um usuário quanto uma empresa.

As políticas são representadas pelo conjunto: [Grupo de Usuários, Ações, Grupo de Recursos e Relacionamento]. Deste conjunto, os Grupos de Usuários e de Recursos são representados por nomes. As Ações devem corresponder a uma ou mais ações pré-definidas. O Relacionamento em uma política deve estar de acordo com um relacionamento definido por alguns objetos no grupo de recursos. Assim, uma política pode ser interpretada como uma autorização de acesso para que qualquer usuário, presente no grupo de usuários, a executar as ações fornecidas em qualquer recurso, presente no grupo de recursos.

Neste modelo, o Gerenciador de Política gerencia as políticas de autorização e executa as verificações de autorização quando invocado. Sua classe possui um método denominado: `isAllowed (Usuário, Ação, Objeto)` que verifica se um usuário tem permissão para executar uma ação sobre o objeto que foi determinado. Ao ser invocado, este método faz primeiro uma busca pelo dono do objeto para se ter acesso às suas políticas de autorização. Para cada política acessada, são feitas as seguintes verificações: i) Se o usuário é realmente membro do grupo de usuários; ii) Se o objeto é membro do grupo de recursos; iii) Se o usuário possui um relacionamento com o objeto. Caso estas verificações não sejam satisfeitas, o Gerenciador de Política retorna uma resposta de negação de acesso ao usuário.



A Figura 5.1 ilustra o funcionamento do *WebSphere Commerce Suíte*, o qual utiliza o esquema desenvolvido por (GOODWIN, GOH & WU, 2002). Neste esquema, o *Controlador de Interação* recebe uma solicitação HTTP de um cliente e chama a *Fábrica de Comando* para selecionar uma implementação do comando apropriado. Este esquema possibilita que o administrador do sistema configure a *Fábrica de Comandos* para selecionar as implementações de comandos mais apropriadas para determinadas situações.

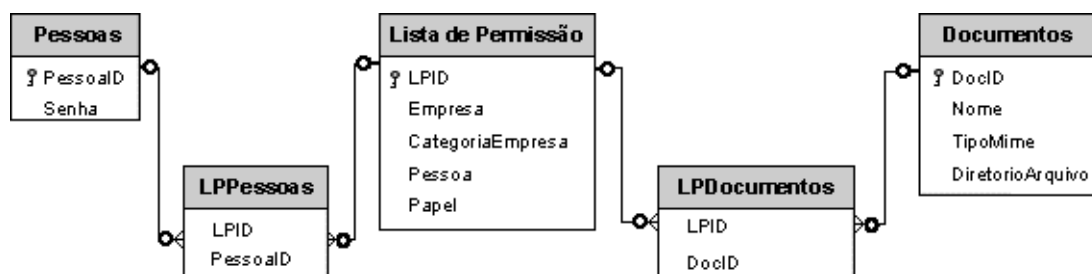


*WebSphere Commerce Suíte*

Através do *Alvo do Comando* são determinados os parâmetros enviados pela solicitação e é solicitada a primeira verificação ao *Gerenciador de Política* para determinar se o usuário tem autorização para executar a ação solicitada. Em caso negativo, o acesso à ação solicitada é negada. Em caso positivo, o comando é executado. Dentro do *Comando*, os *Objetos de Negócio* solicitados pelo sujeito são carregados do Banco de Dados ou, dependendo do perfil do sujeito, do servidor LDAP. Os parâmetros passados para o *Comando* indicam quais instâncias de cada *Objeto de Negócio* o comando deseja operar. No *Comando* também ocorre uma verificação para saber se o usuário pode executar a ação solicitada sobre o *Objeto de Negócio*.

(ROBISON, 2002) também afirma que, apesar do RBAC possuir diversas qualidades, a segurança baseada em papel é inadequada para aplicações típicas de CE B2B. Para justificar este ponto de vista, ele dá o exemplo similar ao apresentado por (GOODWIN, GOH & WU, 2002). No entanto, ele justifica outros problemas que não foram mencionados por (GOODWIN, GOH & WU, 2002). Para (ROBISON, 2002), a geração de papéis em demasia, no intuito de atender às características de alguns sujeitos, gera problemas. Um exemplo disto é a necessidade que o administrador do sistema tem de gerar novos papéis que façam a intersecção dos papéis existentes. No intuito de gerar papéis-intersecção, ocorre um grande crescimento no número de novos papéis, fazendo com que seja cada vez mais difícil gerenciá-los. Para contornar este problema, (ROBISON, 2002) sugere uma técnica denominada: Permissão de Controle de Acesso Baseado em Lista (*permission list-based access control*). Esta técnica possibilita que a segurança seja aplicada a recursos distintos em um sistema B2B.

Para (ROBISON, 2002), em um sistema de CE B2B típico a autorização para se acessar recursos deve ser garantida baseando-se em vários critérios, como por exemplo: a identidade do sujeito, empresa ou papel. Os recursos a serem acessados podem ser: páginas *Web*, documentos no *site*, métodos remotos ou partes de páginas *Web* que mostram informações internas do sistema. Assim, para que seja possível fazer o controle de acesso a estes recursos, é necessário que todos estes componentes sejam armazenados em uma tabela em um Banco de Dados. A Figura 5.2 ilustra as tabelas no Banco de Dados que compõem as informações necessárias para que seja feito o controle de acesso.



Na tabela “Documentos” o campo “nome” representa o nome do documento que é mostrado na página Web como um *hyper-link*, em que os usuários podem clicar para ter acesso ao documento.

A tabela “Lista de Permissão” armazena todas as listas de permissão do sistema. Cada lista de permissão é uma combinação única de: empresa, categoria da empresa (utilizado para classificar as empresas), pessoa e papel. A intenção é que seja feito um & lógico de todos os campos para que seja determinado um nível único de acesso. Para isso, (ROBISON, 2002) dá um exemplo onde há uma lista de permissão com uma empresa com o nome de “*ViewStar*” e um papel com o nome de “Pessoal Executivo”. Os dados nesta lista significam que um sujeito deve ser empregado da empresa *ViewStar* e ter o papel de Pessoal Executivo para que esteja qualificado a exercer esta permissão.

As permissões são relacionadas com os documentos através da tabela “LPDocumentos”. Assim, se este documento está relacionado apenas com a permissão da empresa “*ViewStar*” e papel “Pessoal Executivo”, apenas o sujeito que esteja relacionado com esta permissão tem o direito de acessar este documento. No entanto, caso outras permissões estejam relacionadas com este documento, um sujeito deve atender apenas às exigências de uma permissão para se ter acesso a este documento.

A relação entre um usuário do sistema e as permissões que ele pode ter é estabelecida através da tabela “LPPessoas”.

Através desta técnica, (ROBISON, 2002) acredita que o gerenciamento de permissões se torna mais simples, possibilitando que pessoas sem conhecimento técnico de informática possam gerenciar o conteúdo das configurações de acesso de um sistema de CE B2B.

Mecanismos de Autorização Genéricos para Aplicações Multi-Camadas (*Generic Authorization Mechanisms for Multi-Tier Applications - GAMMA*) é um *framework*

genérico proposto por (ESSMAYR, PROBST & WEIPPL, 2004). Eles afirmam que o objetivo do projeto GAMMA foi desenvolver um *framework* de segurança de alto nível baseado no modelo RBAC, que fosse independente de plataforma e oferecesse mecanismos de segurança como: autenticação, controle de acesso e auditoria. Sua aplicação é destinada a aplicações multi – camadas baseadas em componente. Assim, ele pode ser utilizado no desenvolvimento de sistemas de Comércio Eletrônico *Business-to-Consumer*, *Consumer-to-Consumer* ou *Business-to-Business*.

GAMMA utiliza uma linguagem de definição de segurança (*Security Definition Language - SDL*) que é baseada no padrão XML. Esta linguagem é utilizada para a configuração do *framework* e da política de segurança adotada.

GAMMA é constituído de diversos componentes (Figura 5.5):

1. trata-se do gerenciador central do *framework* onde é feito o controle de todos os componentes de segurança, como: autenticação, controle de acesso ou auditoria. As solicitações feitas por um usuário cliente são recebidas pelo *gerenciador de segurança*, que verifica o tipo de solicitação e a envia para o componente responsável pela mesma.
2. é o componente responsável pela verificação da validade de uma solicitação de acesso. GAMMA permite que mais de um *modelo* de controle de acesso esteja ativo em um dado momento. É possível que sejam utilizados modelos como RBAC ou DAC em uma seqüência pré-determinada. Assim, o *controlador de acesso* recebe do *gerenciador de segurança* todas as solicitações de acesso feitas pelos clientes que, posteriormente as envia para os *modelos* ativos no momento e espera pelo resultado da avaliação. O resultado das avaliações feitas pelos modelos são repassadas para o *gerenciador de segurança* para que sejam feitas as ações correspondentes.
3. este componente é responsável pelo armazenamento de meta informações necessárias para as tomadas de decisão de controle de acesso.
4. trata-se de um modelo de segurança que obtém sujeitos, objetos, autorizações e restrições provenientes do *provedor de dados de segurança* e os transfere para uma *base de regras* através de um *manipulador ACL*. Assim, quando o *controlador de acesso* entra em contato com o *modelo* para a verificação de uma solicitação, inicia-se

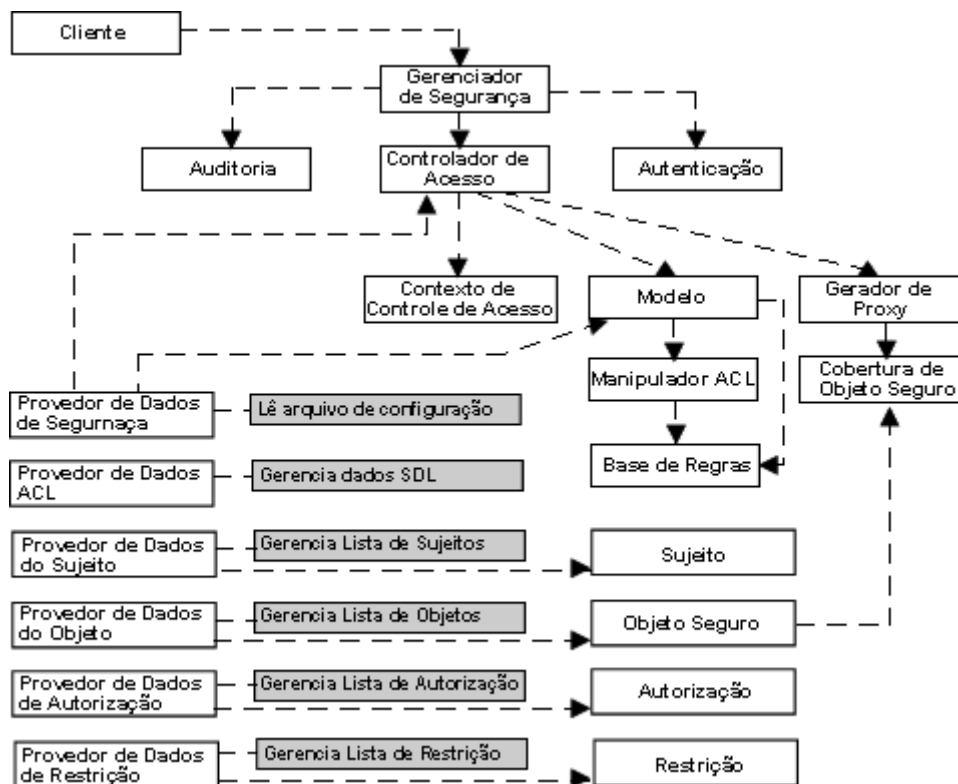
uma busca na *base de regras* enviando como parâmetro o sujeito e o objeto solicitado. Dependendo das regras existentes, são retornados os seguintes valores: i) *true*: se a regra é encontrada e o acesso é garantido; ii) *false*: se a regra é encontrada e o acesso é negado; iii) *weak true*: se nenhuma regra é encontrada e o objeto possui informações irrestritas; iv) *weak false*: se nenhuma regra é encontrada e o objeto possui informações restritas. Caso o resultado retornado seja “forte”, o controlador de acesso para a verificação da solicitação retorna o resultado para o *gerenciador de segurança*. No entanto, se o resultado for “fraco” é feita a verificação em outro modelo, caso exista. Caso nenhum resultado forte seja encontrado, é retornado o primeiro resultado fraco encontrado.

5. este componente tem por finalidade gerenciar as regras contidas da *base de regras*.
6. : é o componente responsável pelo armazenamento de regras que descrevem como um sujeito pode acessar um objeto, classe ou método. Cada modelo possui sua própria base de regras.
7. este componente tem a tarefa de ler o arquivo de configuração que contém descrito: i) os modelos de segurança que estão sendo utilizados no momento; ii) a combinação e o relacionamento entre os modelos; iii) referência para os provedores responsáveis pelo envio de dados seguros. Este arquivo de segurança é escrito pelo administrador de segurança do sistema.
8. este componente é compreendido de usuários ou processos em um sistema. GAMMA leva em consideração o acesso transitivo, onde, por exemplo, um sujeito deseja executar um método em um determinado objeto. No entanto, a execução deste método exige o acesso a um segundo objeto. Neste caso, os atributos do sujeito são armazenados no componente *Contexto de Controle de Acesso* para que eles possam ser verificados pelo controlador de acesso quando for solicitado o acesso ao segundo objeto.
9. este componente é a base para todos os objetos do sistema que precisam ser protegidos. Nenhum sujeito possui acesso direto aos objetos seguros, somente a um componente privilegiado denominado: *Cobertura do Objeto Seguro*.

Esta cobertura permite que sejam feitas verificações de segurança antes que o objeto seja acessado.

10. este é um componente que tem o papel de decidir se um recurso pode ser concedido ou não. Para tanto, cada componente de autorização possui um método de verificação de acesso que é chamado quando se faz a validação de acesso. GAMMA ainda permite que cada modelo utilize diferentes tipos de autorização, permitindo assim uma maior flexibilidade no processo de autorização.
11. este componente possibilita a restrição de certas ações no sistema. GAMMA diferencia as restrições como: i) *Restrições específicas de modelo*, que influenciam apenas ações e tarefas do modelo de segurança específico. ii) *Restrições não-específicas de modelo*, que são independentes do modelo de segurança ativo no momento. Estes tipos de restrições influenciam a aplicação como um todo. De forma similar ao componente de autorização, este componente também possui um método de verificação que é capaz de garantir ou negar acesso. Assim, o acesso de um sujeito a um objeto só é permitido se os componentes de autorização e restrições permitirem.
12. este componente impossibilita o acesso direto a um objeto seguro. O objeto de cobertura tem funções equivalentes a de um *proxy*. Ele possui os mesmos métodos de acesso que os do objeto seguro. Estes métodos instanciam o objeto seguro para que seus métodos correspondentes sejam acessados. No entanto, antes que os métodos dos objetos seguros sejam acessados, é feita uma verificação no Controlador de Acesso para ver se o sujeito que fez a solicitação possui permissão para o acesso.
13. **Proxy** trata-se de um componente central que tem por função gerar as coberturas dos objetos de segurança.

Estes componentes mencionados podem ser visualizados através do diagrama da Figura 5.3. Com exceção do cliente, todos estes componentes estão localizados no lado do servidor.



GAMMA funciona da seguinte forma: para acessar um determinado *Objeto Seguro*, um *Sujeito* faz uma solicitação ao *Controlador de Acesso* com o pedido de autorização necessária para que o acesso seja permitido. Esta solicitação é passada por todos os *Modelos* ativos especificados no arquivo de configuração. Cada *Modelo* faz uma busca pelas regras armazenadas nas *Bases de Regras* correspondentes, retornando uma lista de possíveis autorizações previamente definidas para a combinação sujeito/objeto. Cada uma destas autorizações é verificada através do método de autorização “*checkAccess()*”. Além disso, existem as restrições que podem ser especificadas no controle de acesso de um sujeito a um objeto. Desta forma, o acesso de um sujeito a um objeto só é concedido após os processos de verificação de autorização e restrição. O resultado destas verificações é retornado para o controlador de acesso que, por sua vez, retorna para o gerenciador de segurança.

Segundo (ESSMAYR, PROBST & WEIPPL, 2004), GAMMA oferece benefícios sem precedentes para o desenvolvimento de sistemas de Comércio Eletrônico se comparado

com outras arquiteturas de segurança. Ele afirma que isto se deve ao fato de GAMMA possibilitar a combinação do que há de melhor em diferentes modelos de segurança.

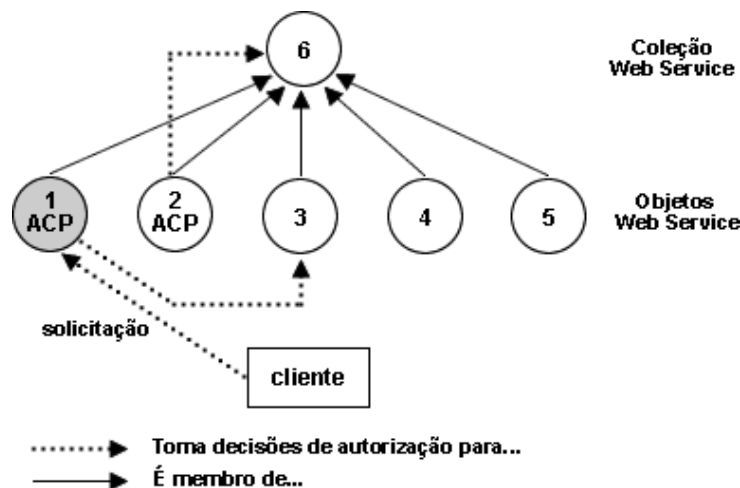
(KRAFT, 2002) realizou um estudo no qual apresentou um modelo abstrato genérico que pode ser utilizado como base no desenvolvimento de um processador de controle de acesso para componentes de *Web Services*. Kraft afirma que o que fundamenta os *Web Services* é que eles possuem um protocolo em comum que possibilita aplicações Web conectarem-se umas às outras através da Internet. Tal característica possibilita a sua utilização na interação entre sistemas de CE B2B. Este modelo é composto por dois tipos de componentes.

O primeiro componente é o *Access Control Processor - ACP* (*Access Controle Processor - ACP*). Ele é um objeto *Web Service* que, juntamente com outro processador de controle de acesso, toma decisões de autorização para um componente *Web Service*. Sendo assim, é possível que um componente *Web Service* possua vários processadores de controle de acesso associados a ele.

O segundo componente é o *Gatekeeper* (*Gatekeeper*). Ele é considerado tanto um processador de controle de acesso, quanto um objeto *Web Service*. A diferença está no fato do porteiro dever tomar a decisão final quando uma solicitação é feita a um componente *Web Service*, decidindo se o acesso deve ser concedido ou negado. Além disso, ele é responsável pela autenticação de usuários que façam solicitações aos componentes. Cada componente deve ter apenas um porteiro associado a ele.

O porteiro intercepta todas as solicitações para um objeto para decidir a permissão de acesso. Ele trabalha junto com um conjunto de processadores de controle de acesso para poder tomar a sua decisão. Este modelo não obriga qualquer restrição particular de implementação. A comunicação pode ser feita em qualquer padrão da Internet como XML, SOAP ou WSDL. Um exemplo deste cenário pode ser visualizado através da Figura 5.4.





Neste caso, um cliente deseja acessar um processo oferecido pelo objeto 3. O porteiro (Objeto 1) intercepta a solicitação e autentica o cliente. Posteriormente ele verifica quais são os ACPs utilizados no controle de acesso do objeto solicitado. Feito isto, o porteiro faz o “direcionamento” da solicitação para o objeto ACP 2 e verifica qual foi a decisão de autorização. A união das decisões tomadas pelos processadores de controle de acesso são analisadas pelo porteiro para então ser tomada a decisão de consentimento ou negação de acesso ao objeto. Posteriormente o porteiro envia uma resposta de volta ao cliente contendo a decisão da autorização.

(KRAFT, 2002) afirma que a solução proposta por sua pesquisa representa um ponto de partida viável que pode ser utilizado como base para futuras pesquisas na área de controle de acesso e segurança para *Web Services*.

Este Capítulo realizou um estudo sobre o estado da arte do controle de acesso em sistemas de CE B2B. Os três primeiros trabalhos descritos não levam em consideração a interação entre os sistemas das empresas parceiras. Já (KRAFT, 2002) considera a interação, mas vincula seu modelo de controle de acesso a um *Web Service*, o que limita o escopo de aplicação nesta abordagem. A partir desta descrição do estado da arte será

descrita no próximo capítulo a aplicação do modelo  $UCON_{ABC}$  em Sistemas de CE B2B que interagem entre si.

Este capítulo apresenta a principal contribuição desta dissertação: a aplicação do modelo  $UCON_{ABC}$  em sistemas de CE B2B que interagem entre si. Primeiramente é feita uma análise da aplicação do  $UCON_{ABC}$  em sistemas de CE B2B. Depois a proposta de aplicação é apresentada ilustrando seus componentes e sua forma de funcionamento. Em seguida é apresentado o Agrupamento Implícito e uma proposta de melhoria desta técnica, de forma que venha a se tornar Agrupamento Implícito Parcial. Esta técnica é incorporada à aplicação proposta para o gerenciamento de permissões. Posteriormente é apresentado o “sujeito composto” um conceito que se originou devido a fatores específicos de sistemas que interagem entre si. Logo após é feita uma comparação entre esta proposta e os trabalhos relacionados. O capítulo é finalizado com uma conclusão sobre a proposta de aplicação.

(PARK & SANDHU, 2004) explicam o funcionamento do  $UCON_{ABC}$  usando, para facilitar a compreensão das novas características do modelo, os sistemas DRM. Neste tipo de sistema um usuário (sujeito) pode ter acesso (autorização) a um conteúdo digital (objeto) desde que possua licença digital (direito) sobre o mesmo. Um exemplo é uma aplicação onde um usuário pode escutar arquivos de áudio desde que possua créditos (atributos do sujeito) o suficiente. Na medida em que este usuário escuta uma música, seus créditos vão sendo diminuídos de acordo com o preço por unidade de tempo (atributo do objeto) até que o usuário não tenha mais direitos sobre o arquivo. Desta forma, fica clara a visualização de conceitos como continuidade dos direitos e mutabilidade de atributos.

A compreensão da aplicação do modelo  $UCON_{ABC}$  em sistemas como o de CE B2B é diferente. Nestes ambientes o usuário (sujeito) deve ter autorização tanto para entrar no sistema como também para executar diferentes ações no sistema. Isto porque um usuário

tem acesso apenas às ações que estão relacionadas ao seu papel, nível ou lista de acesso. Então surge o questionamento: O objeto neste caso seria o sistema como um todo ou cada uma das ações que compõem o sistema? A resposta é: ambos. É necessário que haja dois níveis de autorização: a do sistema e a das ações do sistema. Em ambos os níveis é possível aplicar os conceitos de Autorização, Obrigação, Condição e Mutabilidade. A grande diferença está no conceito de Continuidade. Quando o sistema é visualizado como um objeto é possível aplicar a Continuidade, pois o usuário pode ser constantemente monitorado enquanto exerce seu direito de acesso ao sistema. Ao extrapolar, por exemplo, o limite de ações indevidas dentro do sistema, o direito de acesso deste usuário pode ser interrompido imediatamente. No entanto, quando ações são visualizadas como objetos, não é possível aplicar o conceito de Continuidade. Na solicitação de execução de uma ação, o sistema de controle de acesso verifica se o usuário solicitante possui direitos sobre a ação. Em caso positivo a ação é executada em frações de segundo, ou seja, não há como haver Continuidade se o tempo de acesso e execução de uma ação é muito pequeno.

A proposta de aplicação do modelo  $UCON_{ABC}$  em sistemas B2B tem por objetivo prover o controle de acesso especificamente entre os sistemas das empresas parceiras que interagem entre si. Esta dissertação não trata do controle de acesso que os sistemas fazem para com seus próprios usuários.

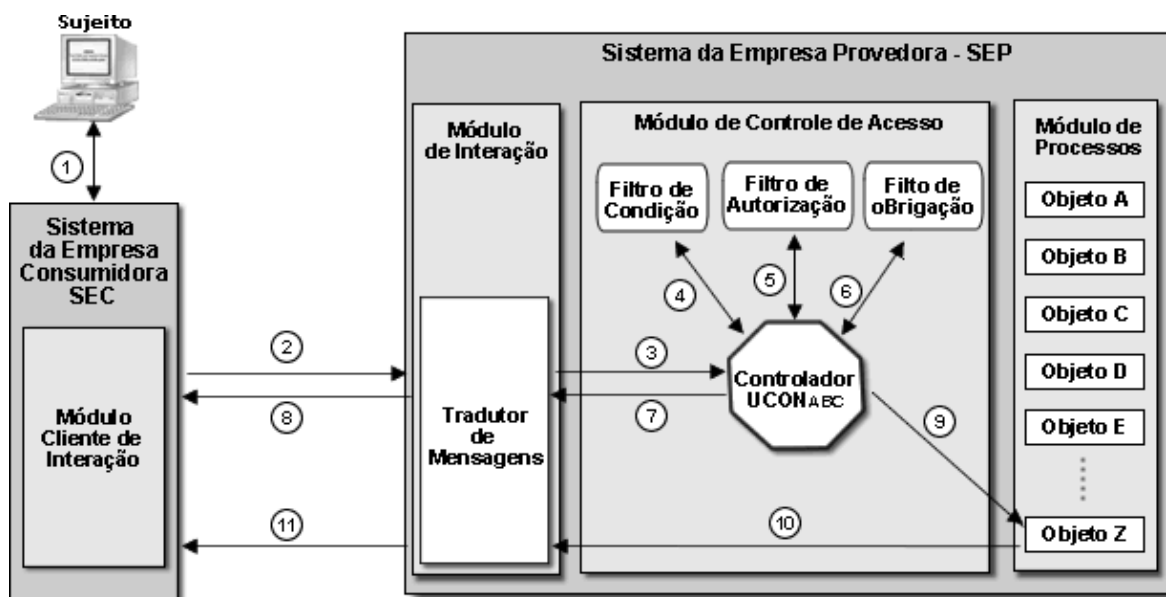
A proposta de aplicação é composta por dois sistemas, os quais são (Figura 6.1): Sistema da Empresa Provedora (SEP) e Sistema da Empresa Consumidora (SEC).

No SEP estão todas as funcionalidades necessárias para o controle de acesso, interação e execução de processos de negócios. Sendo assim, este sistema é constituído de três módulos distintos:

1. O primeiro Módulo é o de *Interação*, que possibilita a interação entre o SEP e os SECs, ou seja, traduz as mensagens transmitidas entre os sistemas.
2. O segundo Módulo é o de *Controle de Acesso*, que é responsável por estabelecer o controle de acesso obedecendo aos conceitos básicos do modelo  $UCON_{ABC}$ .
3. O *Módulo de Processos* compreende os processos de negócios que são funções, rotinas ou métodos implementados e configurados para serem acessados remotamente. Estes processos atendem pelo nome de “Objetos”, como analogia ao modelo  $UCON_{ABC}$ .

O SEC representa os parceiros comerciais que desejam interagir com seu parceiro fornecedor de produtos através de seu próprio sistema. Assim, apesar desta proposta apresentar apenas um SEC, na realidade o SEP pode interagir com muitos outros SECs. Em se tratando especificamente do Sistema de CE B2B, este sistema possui um Módulo Cliente de Interação que realiza a conversação com o SEP. Da mesma forma que no sistema servidor, este módulo pode utilizar uma das opções de ferramentas de interação existentes para poder se comunicar.

A Figura 6.1 ilustra esta proposta de aplicação.



Para explicar a forma de funcionamento da proposta, é apresentado um caso similar ao descrito na seção 4.1, onde uma Empresa Montadora de Computadores (EMC) possui um SEC que interage com um SEP de uma Empresa Fabricante de Processadores (EFP) para transações entre as duas empresa.

De acordo com o contexto acima, um usuário da EMC que deseja trocar informações com a EFP deve, primeiramente, autenticar-se no SEC da EMC. Posteriormente é necessário autenticar-se no SEP da EFP para que a partir de então seja possível a interação entre os sistemas.

No processo de autenticação primeiramente é verificado se o SEC de origem do pedido de autenticação é realmente quem diz ser. Esta verificação pode ser feita simplesmente através da verificação do IP de origem do SEC ou então de maneira mais sofisticada, como por exemplo, através de certificados digitais.

Sem o processo de autenticação de um sistema, um usuário com conhecimentos avançados poderia copiar um SEC e instalá-lo em outro local. Com isto, este sistema “clonado” poderia acessar os serviços de um SEP como se fosse o sistema original. Tal situação poderia causar sérios danos para ambas empresas parceiras.

Caso o resultado do processo de autenticação do SEC no SEP seja positivo, é liberado então o processo de autenticação do usuário propriamente dito. Quando este processo é finalizado sem erros, o sistema de controle de acesso do SEP inicia a busca pelas permissões as quais este usuário tem acesso. Estas permissões devem ser alocadas na memória do sistema para serem utilizadas pelo sistema de controle de acesso posteriormente.

Observando a figura 6.1, quando um usuário da EMC deseja fazer uma consulta de preços dos produtos da EFP ele deve, após autenticar-se no SEP, executar o comando de pedido da lista de produtos (passo 1). O Módulo Cliente de Interação recebe esta solicitação

e se encarrega de encaminhá-la para o SEP através de alguma ferramenta cliente de interação (passo 2).

O Módulo de Interação do SEP recebe esta solicitação e a traduz para a tecnologia utilizada no sistema. A partir deste ponto, a solicitação é enviada para o Controlador UCON<sub>ABC</sub> (passo 3). Qualquer solicitação recebida pelo Módulo de Interação é, obrigatoriamente, direcionada para este Controlador. Sua implementação deve seguir as especificações do modelo UCON<sub>ABC</sub> na qual, através dos atributos do sujeito e do objeto a ser acessado, são verificadas as regras de condição, autorização e obrigação do SEP através de seus respectivos filtros (passos 4, 5 e 6). Estas verificações são coordenadas sequencialmente de forma que, em caso de negação de acesso por algum dos filtros, o Controlador pára de executar as tarefas subseqüentes. O primeiro filtro verificado é o de Condição. Isto porque caso haja alguma irregularidade, por exemplo, no horário de acesso, no endereço IP do SEC do usuário ou no estado do sistema, os demais filtros nem precisam ser verificados. O Filtro de Autorização é verificado em seguida porque se não houver autorização de acesso, não há razão para se verificar o Filtro de Obrigação. É no Filtro de Autorização que são verificadas as permissões do usuário que foram coletadas em seu processo de autenticação. O Filtro de Obrigação se encarrega de verificar se o usuário deve executar alguma ação prévia antes de ter acesso a um determinado objeto.

O conceito de Continuidade desta proposta está no fato de todas as solicitações de acesso a um objeto serem controladas por estes três filtros. Assim, mesmo após autenticar-se o usuário é constantemente monitorado pelo SEP.

Em caso de negação de acesso, o Controlador UCON<sub>ABC</sub> retorna ao Módulo de Interação uma resposta de negação de acesso ou de necessidade de cumprimento de alguma obrigação (passo 7). No Módulo de Interação esta resposta é traduzida em uma mensagem que é retornada ao SEC (passo 8).

Caso a solicitação passe pela verificação de todos os filtros, então o acesso ao objeto desejado é liberado (passo 9). Após serem acessados e executados, os Objetos podem retornar ao Módulo de Interação diferentes tipos de dados, como por exemplo: uma lista de produtos, um arquivo de contrato ou então valores lógicos do tipo verdadeiro/falso (passo 10). Após converter os valores de retorno em uma mensagem, o Módulo de Interação a

envia para o SEC (passo 11) que, por sua vez traduz novamente o conteúdo para o formato utilizado e exibe a resposta para o usuário. Assim, o ciclo de funcionamento do modelo de implementação é finalizado. Quaisquer outras ações de interação entre os sistemas SEC e SEP obedecem este mesmo ciclo.

A proposta da seção 6.2 ilustra o processo de controle de acesso, utilizando as características do  $UCON_{ABC}$ , após o usuário do SEC ter se autenticado no SEP. No entanto, isto não é o bastante. É também necessário definir como deve ser feito o gerenciamento de permissões das empresas e seus usuários em um sistema B2B, seja através de papéis, níveis ou lista de controle de acesso.

Como já mencionado no Capítulo 3, o Modelo  $UCON_{ABC}$  abrange modelos de controle de acesso tradicionais como DAC, MAC ou RBAC. O modelo RBAC ganhou destaque nos últimos anos por possuir características que facilitam o gerenciamento de permissões entre sujeitos e objetos. (SANDHU, 2001) destaca a predominância do RBAC na década de 90 e afirma que sua aplicação em sistemas de CE B2B e B2C seria o principal motivo da predominância do RBAC nesta década.

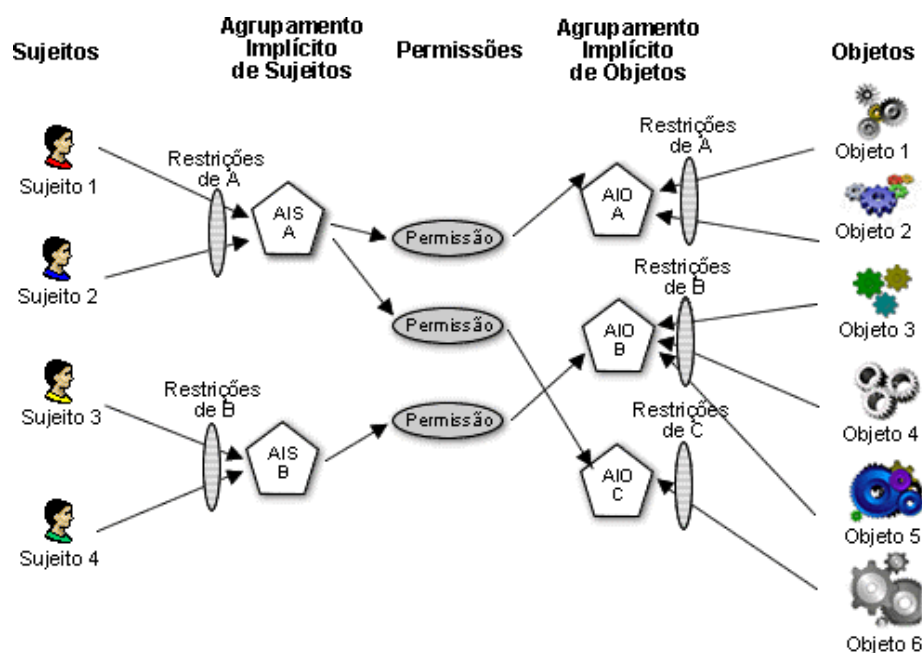
Apesar de (SANDHU, 2001) ter mencionado a utilização do RBAC para sistemas de CE B2B, (ROBISON, 2002) e (GOODWIN, GOH & WU, 2002) afirmam que o RBAC possui limitações quanto a sua utilização neste tipo de sistema e propõem melhorias. (GOODWIN, GOH & WU, 2002) propõem uma técnica que estende o RBAC, denominada “Agrupamento Implícito”, descrita na seção 5.1.

(GOODWIN, GOH & WU, 2002) possuem argumentações convincentes quanto à limitação do RBAC e das vantagens existentes em sua proposta. Sendo assim, foi feito um estudo sobre esta técnica para que ela pudesse ser incorporada à proposta de aplicação do  $UCON_{ABC}$  em sistemas B2B.

Apesar de descrever o funcionamento do Agrupamento Implícito, (GOODWIN, GOH & WU, 2002) não apresentam uma representação gráfica clara de sua técnica, o que



dificulta a sua visualização e compreensão. Sendo assim, outra contribuição desta dissertação é a apresentação de uma representação gráfica desta técnica. A descrição do funcionamento do Agrupamento Implícito já foi feita na seção 5.1. Esta seção fará novamente sua descrição baseando-se, no entanto, na Figura 6.2 que ilustra o Agrupamento Implícito.



O principal motivo para o desenvolvimento desta técnica foi o excesso de criação de papéis que variam, neste tipo de sistema, devido à existência de diferentes funções, empresas e localizações. Com isso, os nomes dos papéis determinariam os sujeitos que um administrador deveria atribuir estes papéis. Assim, estes papéis deixariam de ser genéricos para se tornarem papéis específicos para serem atribuídos a um ou dois sujeitos cada. (ROBISON, 2002) afirma que quanto maior a quantidade de papéis criados, maior a complexidade no gerenciamento de permissões no sistema.

O Agrupamento Implícito tem por objetivo agrupar sujeitos e objetos em grupos distintos de acordo com seus atributos. Diferentemente do RBAC, a permissão entre um sujeito e um objeto não é feita por intermédio de um papel, mas sim por intermédio da permissão entre o grupo que o sujeito pertence e o grupo que o objeto pertence.

Na Figura 6.2 existem dois Agrupamentos Implícitos de Sujeitos (AIS): i) O AIS A, que possui os sujeitos 1 e 2; ii) O AIS B, que possui os sujeitos 3 e 4. Há também três Agrupamentos Implícitos de Objetos (AIO): i) O AIO A, que possui os objetos 1 e 2; ii) O AIO B, que possui os objetos 3, 4 e 5; iii) E o AIO C, que possui o objeto 6. Com isto, se o sujeito 4 deseja acessar o objeto 5, é necessário que seu grupo (AIS B) tenha permissão para acessar os objetos do grupo ao qual pertence o objeto 5 (AIO B). Neste caso, o acesso seria permitido. O acesso só seria negado caso este mesmo sujeito quisesse acessar o objeto 1, pois o grupo a que pertence não possui relação de permissão com o grupo do sujeito.

Para pertencer a um grupo é necessário que o sujeito, ou objeto, satisfaçam algumas restrições obrigatórias pré-estabelecidas. As restrições para um AIS podem ser a empresa a que o sujeito pertence, o país de localização da empresa, o papel que desempenha na empresa, dentre outros. Um exemplo neste caso seria que, para pertencer ao AIS A, um sujeito deve ser funcionário da empresa *BestComp*, que esteja localizada no *Brasil*, e que possua papel de *Negociador*. As restrições para um AIO podem ser: tipo de objeto, tipo de execução, dentre outros. Um exemplo neste caso seria que, para pertencer ao AIO B, um objeto deve ser do tipo *Negociação*, onde seja possível apenas *Ler*.

O Agrupamento Implícito possui duas grandes vantagens sobre o RBAC no que diz respeito à administração de permissões. A primeira delas é o fato de que se um atributo do sujeito é alterado, como por exemplo, o país em que trabalha, a sua mudança de grupo será automática, sem que haja a intervenção do administrador do sistema. A segunda vantagem refere-se ao caso em que haja alguma outra restrição que deva ser levada em consideração para que um sujeito ou objeto faça parte de um grupo. Utilizando o Agrupamento Implícito, basta definir um novo grupo e atribuí-lo às permissões necessárias. Utilizando o RBAC, seria necessário definir novos papéis e então reatribuí-los a todos os sujeitos que estejam com os papéis antigos, o que acarretaria em um maior custo de administração.

Estas duas vantagens se devem ao fato de no Agrupamento Implícito não haver a necessidade de relacionar diretamente os usuários com os seus grupos<sup>1</sup>. O próprio sistema

---

<sup>1</sup> Muitos sistemas implementam esta relação através de tabelas de Banco de Dados ou arquivos XML, como é o caso do XACML.

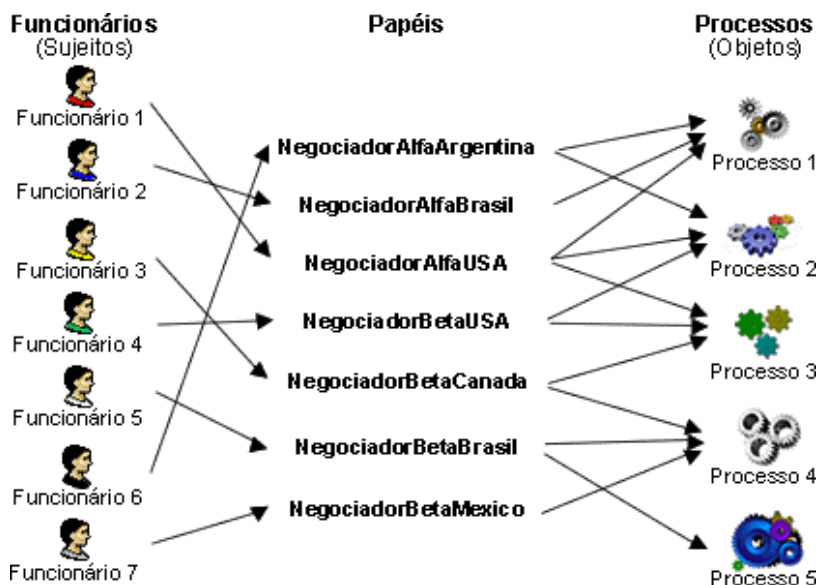
de controle de acesso se encarrega de fazer isso, através dos atributos do sujeito, sem a intervenção de um administrador.

Apesar das vantagens mencionadas, (GOODWIN, GOH & WU, 2002) afirmam que além de agrupar os sujeitos, é necessário agrupar os objetos a serem acessados. Apesar deles afirmarem que esta característica é uma vantagem, ela possui uma maior complexidade na gestão sobre estes grupos de objetos.

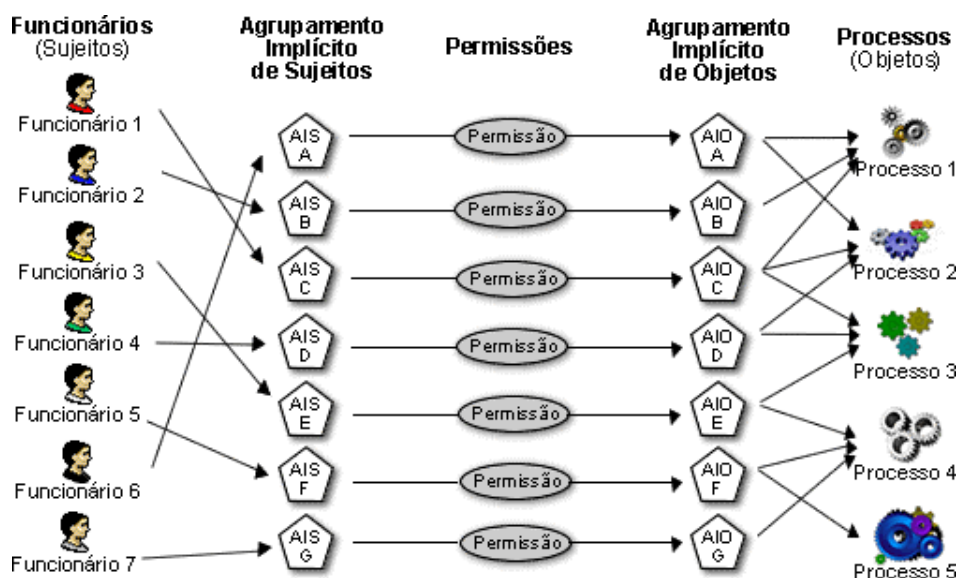
Para que esta observação fique clara, será considerado e comparado um caso hipotético entre o modelo RBAC e o Agrupamento Implícito. Tomando como exemplo funcionários (sujeitos) de duas empresas multinacionais diferentes, denominadas Alfa e Beta, que possuem SEC que interagem com um SEP de uma empresa fornecedora de produtos. Os funcionários analisados são todos negociadores e trabalham nas filiais das empresas, localizadas em diferentes países. Os processos (objetos) a serem acessados no SEP variam de acordo com o funcionário de cada filial. Desta forma, é necessário que estes funcionários possuam permissões específicas de acordo com a empresa, a função e o local de trabalho em que se encontram. A relação entre os funcionários, seus atributos e os processos a serem acessados pode ser visualizada através da Tabela 6.1.

<b>Funcionários</b>	<b>Empresa</b>	<b>País</b>	<b>Função</b>	<b>Serviços a serem acessados</b>
funcionário 1	Alfa	USA	Negociador	1, 2 e 3
funcionário 2	Alfa	Brasil	Negociador	1
funcionário 3	Beta	Canada	Negociador	3 e 4
funcionário 4	Beta	USA	Negociador	2 e 3
funcionário 5	Beta	Brasil	Negociador	4 e 5
funcionário 6	Alfa	Argentina	Negociador	1 e 2
funcionário 7	Beta	Mexico	Negociador	4

Caso o sistema de gerenciamento de permissões fosse implementado tendo como base o modelo RBAC, a relação entre os funcionários, os papéis e os objetos a serem acessados poderiam ser visualizados através de Figura 6.3.



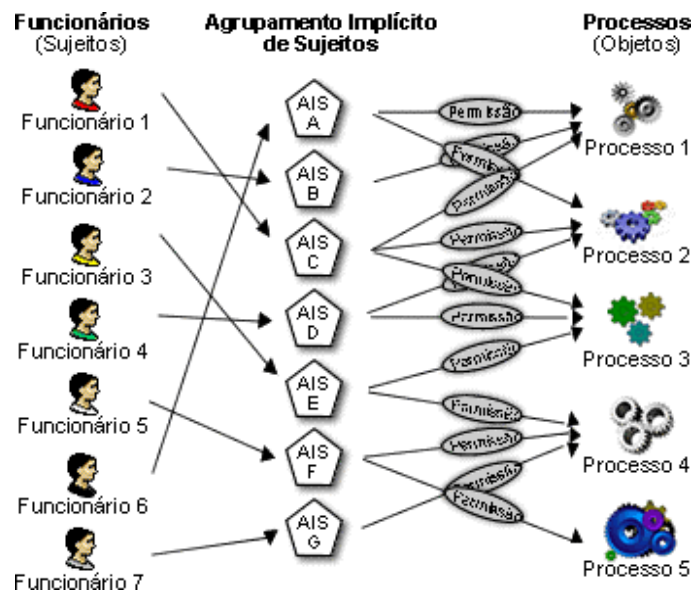
Ao observar a Figura 6.3 é possível perceber que para atender a todos os atributos dos funcionários e os processos a que eles têm direito, foi necessário criar sete papéis distintos. No entanto, se o Agrupamento Implícito for aplicado a este mesmo exemplo, a relação entre funcionários e processos pode ser visualizado através da Figura 6.4.



Através da Figura 6.4, observa-se que os papéis foram substituídos pelos AIS e foram criados AIOs que aglomeram os objetos que devam ser acessados pelo grupo de AIS.

Este exemplo foi propositalmente gerado de forma que houvesse uma grande variação entre os objetos que podem ser acessados por cada AIO. Em piores casos como este não há outra escolha a não ser formar, por exemplo, um AIO que atenda especificamente a um AIS. Assim, o gerenciamento de um Agrupamento Implícito pode ser uma tarefa muito complexa, pois (GOODWIN, GOH & WU, 2002) afirmam que os objetos devem ser agrupados de acordo com os seus atributos. Isto ocasionaria um custo desnecessário e complexo de gestão, pois da mesma forma que pode haver vários AISs, também podem haver vários AIOs. Com a necessidade de um AIO para cada AIS, a existência de um AIO torna-se desnecessária, pois é perfeitamente possível relacionar os AISs diretamente aos processos de que tem direito. Apesar de ser considerado um “pior caso”, exemplos como este não são difíceis de acontecer devido a diversos fatores de variação entre diferentes funções que funcionários de diferentes empresas e países podem executar.

No intuito de solucionar o problema descrito na seção 6.2.3.2, esta dissertação propõe Agrupamento Implícito Parcial, na qual apenas os sujeitos devam ser agrupados. Desta forma, as permissões seriam atribuídas diretamente aos AISs, como mostra a Figura 6.5.



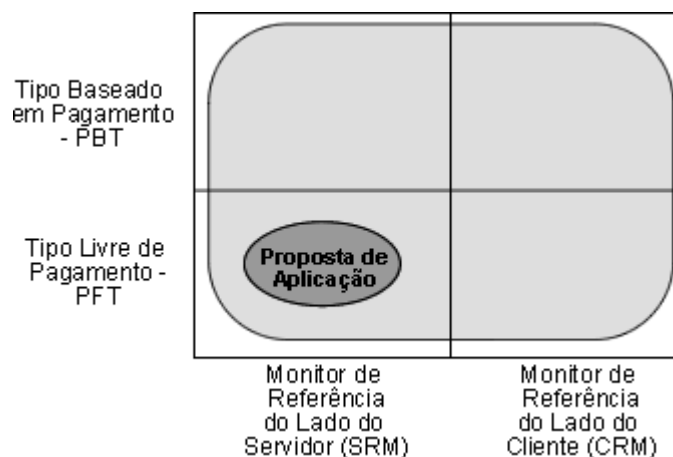
Ao se observar a Figura 6.5 da proposta feita pode surgir um questionamento: Esta proposta não equivale exatamente ao modelo RBAC? A resposta é: aparentemente sim. No entanto, é importante lembrar que na prática este modelo ainda possui as vantagens de facilidade de gestão de usuários e grupos, descritos na seção 6.2.3.1 com a única diferença da exclusão dos AIOs. Com isto, o AIP torna-se mais simples que o AI, facilitando assim a gestão de permissões.

Além de ser utilizado para o gerenciamento de permissões o Agrupamento Implícito Parcial deve ser utilizado pelo sistema para realizar verificações de autorização de um sujeito a um objeto. Os algoritmos que implementam estes processos de verificação devem estar localizados no Filtro de Autorização (Figura 6.1), que é instanciado toda vez que o Controlador UCON recebe uma solicitação de um sujeito para acessar um objeto.

(PARK, 2003) definiu oito arquiteturas UCON que consideravam tanto a localização do Monitor de Referência, quanto o Tipo de Pagamento envolvido. A proposta de aplicação do UCON<sub>ABC</sub> em sistemas B2B considera as arquiteturas SRM e PFT (seção 3.3). A arquitetura SRM leva em consideração a localização do Monitor de Referência no lado do servidor, sendo representado pelo SEP que controla o acesso das empresas parceiras aos seus processos disponibilizados.

A arquitetura PFT leva em consideração a troca segura de informações sem que seja necessária a realização de transações financeiras. O controle de acesso em sistemas de CE B2B não está diretamente relacionado com transações financeiras, como é o caso de sistemas DRM. No entanto, o controle de acesso é um requisito de segurança essencial, pois são compartilhadas informações sigilosas como: valores de produtos, contratos ou propostas para estabelecimento de negócios. Assim, a proposta desta dissertação baseia-se na arquitetura PFT por não levar em consideração o pagamento. A Figura 6.6 ilustra a

localização da proposta de aplicação perante as arquiteturas mencionadas por (PARK, 2003).



Existe uma particularidade no controle de acesso quando se trata de sistemas de CE B2B que interagem entre si. A interação permite a troca de informações entre dois sistemas de empresas parceiras. Como foi visto na seção 6.2.2 um usuário deve autenticar-se primeiro no sistema da empresa em que trabalha para só então poder autenticar-se no sistema da empresa parceira. Além disso, também foi descrita a importância do processo de autenticação entre um SEC e um SEP.

Devido a este contexto surge outra questão: O sujeito que tem acesso ao SEP, ou aos seus objetos, é o SEC ou o usuário do SEC? A resposta é: ambos. Tanto o usuário do SEC, quanto o próprio SEC são sujeitos do SEP, pois ambos passam por um processo de autenticação. Além disso, o sistema de controle de acesso leva em consideração os atributos do sujeito (função, empresa ou filial) e os atributos do SEC (endereço IP ou certificado digital). Desta forma, em sistemas como o de CE B2B um sujeito deve ser considerado como um “sujeito composto”.

A proposta de aplicação do  $UCON_{ABC}$  em sistemas de CE B2B que interagem entre si, descrita nas seções anteriores, possibilita que o controle de acesso seja feito sobre este sujeito composto através dos Filtros de Autorização, Obrigação e Condição.

Na pesquisa feita sobre os trabalhos relacionados (capítulo 5), foram encontradas propostas de modelo, esquema, *framework* ou técnica que melhor atendesse às necessidades do controle de acesso em sistemas de CE B2B.

(ROBISON, 2002), por exemplo, sugere uma técnica simples denominada: Permissão de Controle de Acesso Baseado em Lista. Esta técnica se baseia no modelo RBAC e o conceito da relação dos papéis com as listas de acesso melhora a organização dos dados no controle de acesso em sistemas B2B. Esta técnica visa apenas fornecer o controle de acesso em um ambiente central em que tanto os usuários da empresa fornecedora, quanto os usuários da empresa parceira acessam o mesmo sistema.

(GOODWIN, GOH & WU, 2002) definiram um esquema para sistemas de *E-marketplace*, denominado Controle de Acesso Baseado em Política, que se baseia no modelo RBAC, mas que possui algumas melhorias que, segundo eles, tornam o controle de acesso mais conciso e eficiente. Para tanto, eles definiram o já descrito método denominado “Agrupamento Implícito”. Seu objetivo foi definir um esquema de controle de acesso para sistemas de *E-marketplace*, onde todas as transações são feitas em um sistema central.

(ESSMAYR, PROBST & WEIPPL, 2004) propõem um *framework* genérico denominado GAMMA. Por ser genérico, (ESSMAYR, PROBST & WEIPPL, 2004) afirmam que GAMMA pode ser aplicado a diferentes tipos de sistemas de CE como: *Business-to-Consumer* (B2C), *Consumer-to-Consumer* (C2C) ou até mesmo *Business-to-Business* (B2B). GAMMA também se baseia no modelo RBAC. Além disso, ele é um *framework* independente de plataforma direcionado para aplicações de multicamadas baseadas em componentes, oferecendo mecanismos de segurança como: autenticação, controle de acesso e auditoria.



(KRAFT, 2002) desenvolveu um estudo em que define um modelo geral abstrato, específico para componentes de *Web service*, que pode ser utilizado como base no desenvolvimento de um processador de controle de acesso a sistemas de CE.

Os três primeiros trabalhos aqui mencionados fazem referência a sistemas de CE B2B que possuem um sistema central que é acessado diretamente por usuários de empresas parceiras. Nestes casos a solução de controle de acesso não leva em consideração a interação entre os sistemas das empresas parceiras. A solução proposta por (KRAFT, 2002) vincula um modelo de controle de acesso a uma ferramenta específica: *Web Service*. A pesquisa desta dissertação diferencia-se das demais por definir uma forma de aplicação do  $UCON_{ABC}$  em Sistemas de CE B2B que interagem entre si, independentemente da tecnologia de interação utilizada.

Este capítulo apresentou uma proposta aplicação do  $UCON_{ABC}$  em sistemas B2B que interagem entre si. A junção desta proposta com o Agrupamento Implícito Parcial completam uma forma de implementação ideal para sistemas de CE B2B. Isto porque une conceitos do modelo  $UCON_{ABC}$ , interação entre sistemas, e uma solução de gerenciamento de permissões adequada para sistemas de CE B2B. O Capítulo 7 valida a proposta deste capítulo através da descrição da implementação do protótipo de um sistema de CE B2B que utiliza esta proposta.

Este capítulo apresenta uma descrição tanto das tecnologias utilizadas, quanto da implementação da proposta do modelo de implementação do modelo UCON<sub>ABC</sub> em sistemas de CE que interagem entre si.

Para o desenvolvimento do sistema proposto, foram escolhidas tecnologias livres, porém conceituadas no meio comercial e acadêmico.

Segundo (NEWMAN, 1997), Java é uma linguagem de programação orientada a objetos desenvolvida pela Sun Microsystems em meados da década de 90. A Sun agrupou as tecnologias Java em três edições distintas: i) *Plataforma Java 2 Micro Edition* (J2ME™), utilizada no desenvolvimento para dispositivos móveis e de baixo poder de processamento como *palm tops* ou telefones celulares. ii) *Plataforma Java 2 Standard Edition* (J2SE™), utilizada no desenvolvimento de aplicações locais. iii) *Plataforma Java 2 Enterprise Edition* (J2EE™), utilizada no desenvolvimento de sistemas em rede localizados no lado do servidor (HARDEE).

Esta dissertação fez uso da plataforma J2EE, pois foi desenvolvido um sistema de CE B2B, localizado no lado do servidor.

Segundo (BODOFF, 2003), a tecnologia J2EE suporta uma variedade de tipos de aplicações, abrangendo desde aplicações web de grande porte até pequenas aplicações cliente/servidor. Ela utiliza um modelo de aplicação distribuída multicamada, dividindo a lógica da aplicação em componentes de acordo com sua função. Um componente J2EE é uma unidade de software funcional independente que faz parte de uma aplicação J2EE e

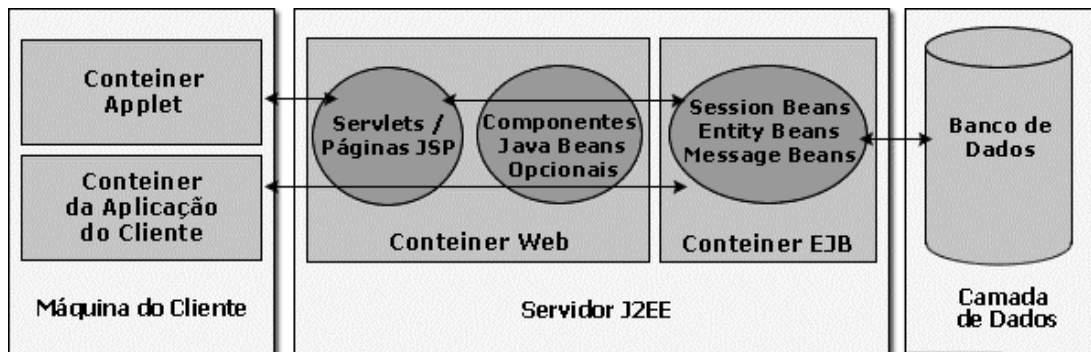
permite realizar comunicações com outros componentes relacionados. A especificação J2EE define os seguintes tipos de componentes:

- *Applets*, que são componentes executados no lado do cliente.
- *Servlets* e *JavaServer Pages (JSP)* são componentes Web que são executados no lado do servidor. Um *Servlet* é uma classe Java usada para estender as capacidades dos servidores que hospedam aplicações acessadas via um modelo de programação do tipo solicitação-resposta. A tecnologia *Java Server Pages (JSP)* permite colocar fragmentos de código do *Servlet* diretamente em um documento baseado em texto, como uma página HTML.
- *Enterprise Java Beans (EJB)* são componentes de negócio que também são executados no lado do servidor. *EJB* encapsula a lógica de negócios de uma aplicação. A lógica de negócio é o código que satisfaz o objetivo da aplicação. A tecnologia *EJB* permite ao desenvolvedor se preocupar apenas com a lógica de negócios, sem ter de se preocupar em gerenciar detalhes como transação de processos, segurança, balanceamento de carga, *pooling* de conexões ou problemas de performance.

A linguagem Java foi utilizada no desenvolvimento tanto do SEC quanto do SEP. A plataforma J2EE foi escolhida para ser utilizada pelo fato de ambos os sistemas possuírem uma arquitetura cliente/servidor em que a interação com o usuário foi feita através de interface Web. Para prover este tipo de interface, foram utilizadas páginas JSP (*Java Server Pages*) e *Servlets*. Pelo fato do SEP fornecer processos a diversos SEC de empresas parceiras, sua arquitetura foi modelada de forma que os processos oferecidos pudessem atender a uma grande quantidade de interações simultâneas. Para tanto, foi utilizada a tecnologia *EJB (Enterprise Java Beans)*, de forma que suas características como transação de processos, segurança ou balanceamento de carga, tornassem o sistema robusto.

Normalmente, aplicações multicamada possuem um nível de complexidade alto em seu desenvolvimento devido a problemas como: controle de transação, gerenciamento de estado, multiencadeamento, dentre outros detalhes complexos de baixo nível. Um *Container J2EE* já possui todos estes processos, deixando assim o desenvolvedor livre para

se concentrar na resolução dos problemas de negócios que surgirem. Desta forma, os *Containers* são servidores onde são executadas as aplicações J2EE. Existem *Containers* específicos para a gerência de execução de EJBs, como também para gerenciar a execução de páginas JSP e Servlets em aplicações Web. Além destes dois tipos, existem os *Containers* de aplicações cliente e applets que residem na máquina do cliente e não fazem parte dos Servidores J2EE. Esta arquitetura pode ser visualizada através da Figura 7.1.



O sistema desenvolvido nesta dissertação utilizou dois *Containers* distintos. O SEP foi desenvolvido utilizando-se o *Container* JBoss 3.2.6 para execução dos EJBs, na camada de negócios, juntamente com o *Container* Apache Tomcat 5.0.28 para a execução das páginas JSP e Servlets, na camada web. Neste caso, o Tomcat já vem integrado ao JBoss de forma a tornar transparente ao desenvolvedor as diferenças dos *Containers* para as camadas web e de negócios.

O SEC, desenvolvido com os componentes JSP/Servlets, também utiliza o *Container* Apache Tomcat 5.0.28. No entanto, este não possui vínculo algum com o JBoss, pois neste sistema não há a presença de EJBs.

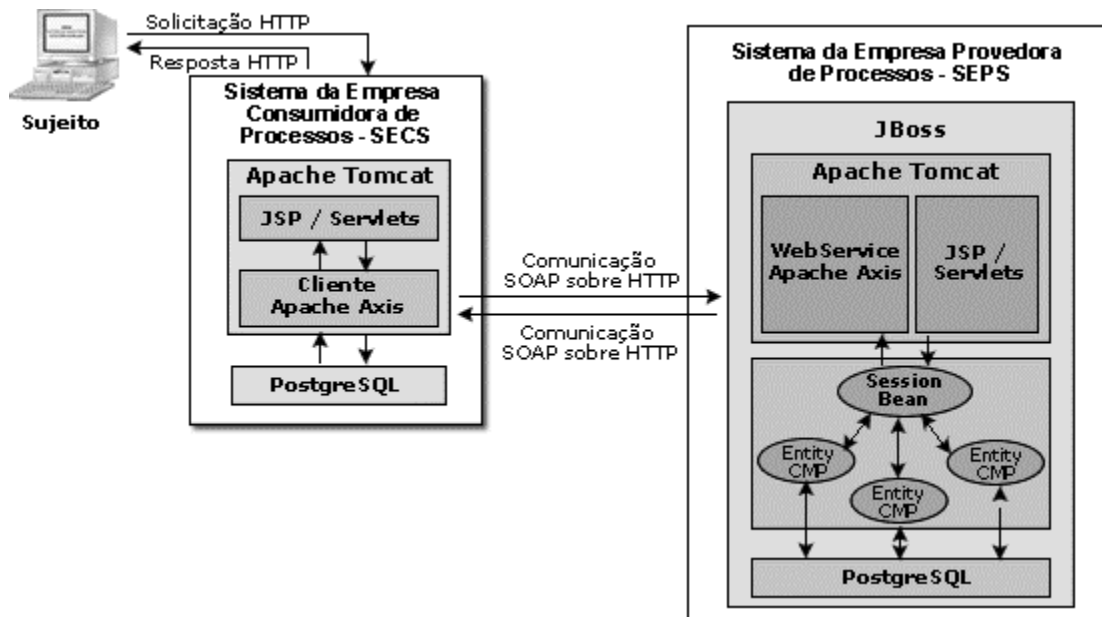
Jboss é um servidor de aplicações J2EE, desenvolvido pelo JBoss Group, que executa e gerencia *Enterprise JavaBeans* (EJB). Ele é baseado em Java, livre e *open source*, podendo ser utilizado em qualquer Sistema Operacional que tenha suporte a Java (TAYLOR, 2004).

Apache Tomcat é um *Container Servlet Open Source* que é utilizado na referência oficial de implementação para as tecnologias Java Servlets e Java Server Pages (JSP). Ele é desenvolvido em um ambiente aberto e participativo, no qual programadores do mundo inteiro podem contribuir para o seu desenvolvimento. Da mesma forma que o JBoss, o Tomcat também é escrito em Java e pode ser utilizado em qualquer sistema operacional (TOMCAT, 2003).

Para prover a interação entre os sistemas, foi escolhido como ferramenta o *Web service* Apache Axis, que é terceira geração do Apache SOAP. Segundo (HOGH et All, 2004), a utilização de *Web services* no desenvolvimento de Sistemas B2B possibilita uma variedade de benefícios como: flexibilidade, reusabilidade, segurança e performance. Além disso, eles também afirmam que Apache Axis é utilizado como meio de interação na aplicação servidora IBM WebSphere, um dos Sistemas B2B mais conceituados no mundo. Axis possui diversas ferramentas que, além de otimizar o processo de desenvolvimento da interação, também auxiliam no controle de acesso.

PostgreSQL foi utilizado, como base de dados, para desenvolver tanto o sistema consumidor quanto fornecedor. Trata-se de um Sistema de Administração de Banco de Dados Objeto-Relacional (ORDBMS) de grande porte, livre, desenvolvido pela Universidade da Califórnia no Departamento de Ciências da Computação de Berkeley. Ele suporta SQL92, SQL99 e oferece muitas outras características como: *queries* complexas, chaves estrangeiras, *triggers*, *views*, integridade de transações e controle de concorrência multi-versões (POSTGRESQL, 2002).

A Figura 7.2 ilustra as tecnologias utilizadas no SEP e SEC:



Esta seção apresenta o protótipo de implementação do sistema de CE B2B. Serão descritos os modelos  $UCON_{ABC}$  utilizados, o banco de dados, filtros e algumas telas do sistema.

Para que o controle de acesso de qualquer sistema seja implementado de acordo com o  $UCON_{ABC}$  não é necessário, e nem viável, que sejam utilizados todos os dezesseis modelos descritos na seção 3.2.2. Para a implementação do controle de acesso do SEP, foram utilizados seis modelos. Estes modelos foram escolhidos levando em consideração que o objeto de acesso é o sistema como um todo, como descrito na seção 6.1.

Este foi o primeiro modelo utilizado. O usuário que utiliza o SEC deve autenticar-se para poder ter acesso aos processos do SEP. Um dos atributos deste usuário é a quantidade

de erros de autenticação. Para cada falha em sua autenticação, este atributo é incrementado. Ao atingir um limite máximo pré-estabelecido, este usuário tem sua conta no SEP, temporariamente desativada.

O segundo modelo é utilizado após o usuário autenticar-se e enquanto estiver utilizando os processos do SEP, ele é monitorado em suas ações. Caso ele faça uma solicitação de um processo ao qual não tem direito, o SEP considera essa solicitação como uma tentativa de ataque. Assim, todo usuário possui um atributo relacionado a erros de acesso não-autorizado. Em cada erro de acesso não-autorizado, este atributo é incrementado. Ao atingir um limite máximo pré-estabelecido, este usuário tem sua conta no SEP temporariamente desativada. Neste caso, considerando o processo como um objeto, o modelo a ser utilizado seria o  $UCON_{preA1}$ .

Na utilização deste modelo, todo usuário do SEC que tenha acabado de ser cadastrado no SEP deve ler um termo de compromisso, aceitá-lo, para então poder utilizar seus processos. Sendo assim, todo usuário possui um atributo que indica sua aceitação do termo de compromisso. Em caso de aceite do termo, este atributo é alterado e o acesso liberado.

Para este modelo, o usuário deve ter duas senhas para utilizar o SEP: a senha de autenticação para acesso ao sistema e a senha crítica. A senha crítica é utilizada para acessar processos críticos com grandes possibilidades de danos para as empresas parceiras, como por exemplo: anulação de pedidos ou cancelamento de contratos. Todo processo considerado crítico deve obrigar o usuário a fornecer sua senha crítica. Desta forma, o atributo do objeto é alterado para que sua execução seja liberada. O atributo do sujeito também é alterado caso o mesmo erre no fornecimento de sua senha crítica, desencadeando assim um processo similar ao descrito para o modelo  $UCON_{preA1}$ . Neste caso, considerando o processo como um objeto, o modelo a ser utilizado seria o  $UCON_{preB1}$ .

Neste modelo o processo de autenticação para acessar os processos do SEP analisa três atributos de condição do ambiente: i) *IP de origem*: é verificado se o IP de origem do pedido de autenticação é o mesmo que o IP do SEC cadastrado no Banco de Dados. É importante salientar que o IP em questão não é do *host* do usuário que está solicitando o pedido, mas sim o IP do SEC utilizado por este usuário. ii) *Horário*: o horário de acesso no SEP pode ser previamente configurado para que qualquer usuário de um determinado SEC possa autenticar-se apenas nos horários devidos. iii) *Data*: cada função exercida por um usuário deve ser configurada de forma que a mesma tenha uma data de início e uma data de término. Sendo assim, caso a data atual em que o usuário estiver solicitado acesso ao sistema estiver fora das datas configuradas, o usuário não mais exerce a função em questão. Como consequência, este atributo pode fazer com que o usuário não mais pertença ao grupo que pertencia anteriormente. Sendo assim, ele perde o acesso aos processos relacionados ao seu antigo grupo.

Por fim, neste último modelo as condições de ambiente citadas para o modelo  $U\text{CON}_{\text{preC0}}$  também se aplicam após o usuário autenticar-se no SEP. Desta forma, após o usuário ter se autenticado, todas as vezes que o mesmo fizer a solicitação de um processo, estas condições serão verificadas para então o acesso ser liberado. Neste caso, considerando o processo como um objeto, o modelo a ser utilizado seria o  $U\text{CON}_{\text{preC0}}$ .

Existem diferentes maneiras de se armazenar as informações necessárias para fornecer o controle de acesso em sistemas computacionais, como por exemplo, em arquivos XML (XACML) ou então em tabelas de Banco de Dados.

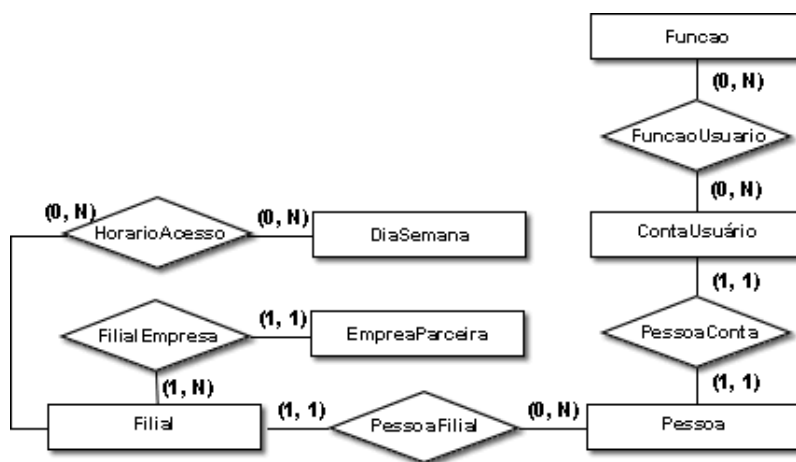
Para esta dissertação foram utilizadas tabelas de um Banco de Dados Relacional, pois na proposta do capítulo 6 existem fatores como os atributos dos sujeitos ou objetos que influenciam na decisão de controle de acesso. A gerência desta grande quantidade de



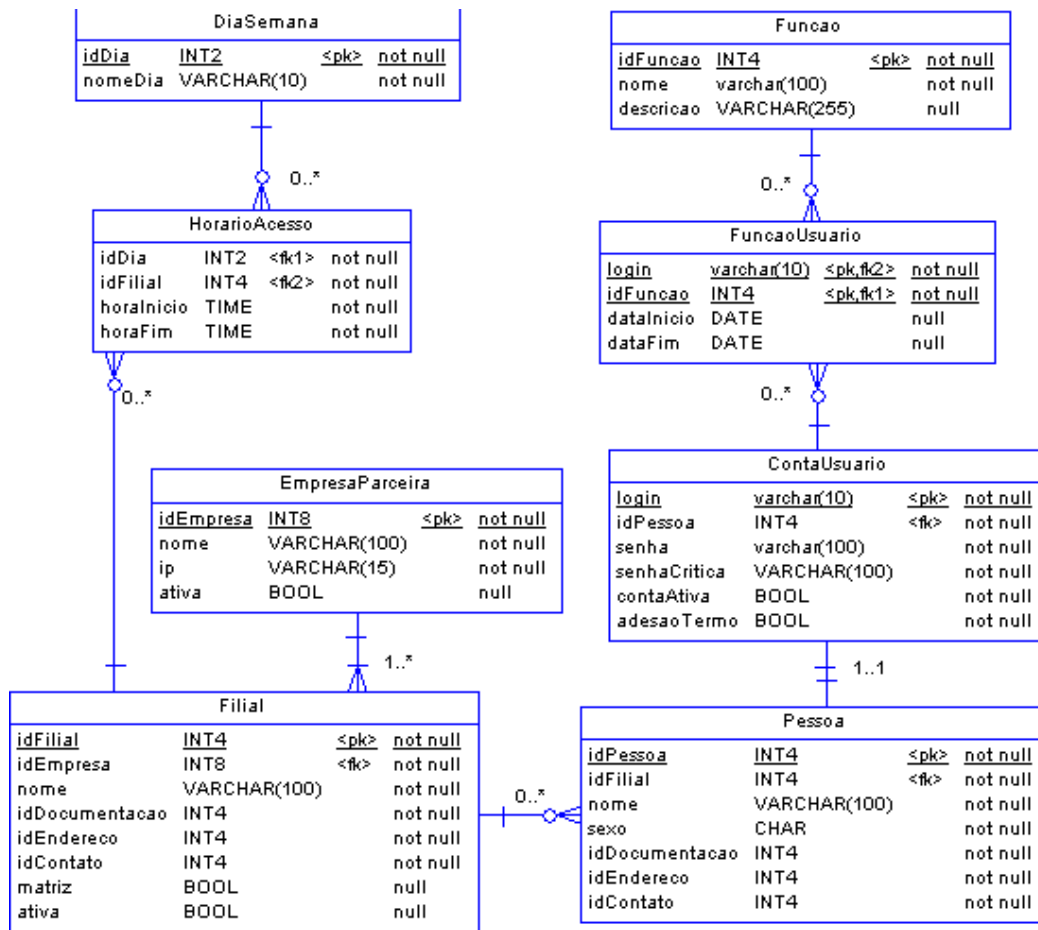
informação em arquivos XML traria uma complexidade de desenvolvimento maior se comparado à utilização de Banco de Dados.

As informações do esquema do Banco de Dados que serão exibidas nesta dissertação referem-se apenas às tabelas necessárias para o armazenamento dos dados relacionados ao controle de acesso. Ainda assim, serão descritas apenas as informações necessárias para a compreensão de cada tabela. As demais tabelas não serão mencionadas. Existem dois grupos distintos de tabelas no esquema do Banco de Dados, as Tabelas de Atributos dos Usuários e as Tabelas de Definição de AISs.

Nestas tabelas encontram-se a maioria dos atributos do usuário para que o sistema possa definir os AISs a que este usuário pertence. A Figura 7.3 ilustra o Diagrama de Entidade e Relacionamento – DER, das tabelas que representam os atributos dos usuários.



A Figura 7.4 ilustra o esquema lógico das tabelas.



esta tabela armazena os dados referentes a conta de um usuário no SEP. No atributo *senha* é armazenada a senha de cada usuário para acesso ao SEP. Este é um atributo utilizado pelo modelo UCON<sub>preA1</sub> (seção 7.2.1.1). No atributo *senhaCritica* é armazenada a senha crítica de cada usuário para o acesso aos processos críticos do SEP. Este é um atributo utilizado pelo modelo UCON<sub>onB2</sub> (seção 7.2.1.4). O atributo de tipo booleano *contaAtiva* é utilizado para determinar se a conta de um usuário está ativada ou não. Em caso positivo o acesso do usuário ao sistema é liberado, senão o acesso é negado. Este é um atributo utilizado pelo modelo UCON<sub>preA1</sub> (seção 7.2.1.1). Por fim, o atributo de tipo booleano *adesaoTermo* é utilizado para determinar se o usuário já aceitou o termo de adesão ou não. Em caso positivo o acesso do usuário ao sistema é

liberado, senão o acesso é negado até que o mesmo aceite o termo de adesão. Este é um atributo utilizado pelo modelo  $UCON_{preB1}$  (seção 7.2.1.3).

esta tabela armazena os dados que identificam a função que o usuário desempenha no SEP. As funções são previamente cadastradas no SEP. Assim, para o cadastro do usuário basta escolher em uma listagem a função que o usuário em questão irá desempenhar.

esta tabela representa um relacionamento, entre as tabelas “ContaUsuario” e “Funcao”, com cardinalidade de muitos-para-muitos. É através dela que são atribuídas as funções que um usuário desempenha dentro do SEP. Através das datas de início e término, é possível verificar se a data atual está dentro do prazo especificado. Caso a data atual seja menor que a data de início ou maior que a data de término, o usuário não mais possui a função em questão. Estas duas datas são atributos utilizados pelos modelos  $UCON_{preC0}$  e  $UCON_{onC0}$  (seções 7.2.1.5 e 7.2.1.6).

esta armazena os dados pessoais do usuário.

esta tabela armazena os dados das filiais em que uma empresa parceira pode ter. Caso a empresa não possua filiais, esta tabela ainda deve ser preenchida. O atributo do tipo booleano *ativa* é utilizado para determinar se a empresa filial está ativada ou não. Em caso negativo, a autorização de acesso de todos os seus usuários ao sistema é negada. Este é um atributo utilizado pelo modelo  $UCON_{preA1}$  (seção 7.2.1.1).

esta tabela armazena os dados comuns da empresa, sendo possuidora ou não de empresas filiais. O atributo *ip* armazena o endereço IP do SEC da empresa parceira que esta interagindo com o SEP. Se a solicitação do usuário tiver como IP de origem o mesmo que o armazenado nesta tabela, então o acesso é liberado, senão é barrado. Este é um atributo utilizado pelos modelos  $UCON_{preC0}$  e  $UCON_{onC0}$  (seções 7.2.1.5 e 7.2.1.6). O atributo do tipo booleano *ativa* determina se a empresa parceira está

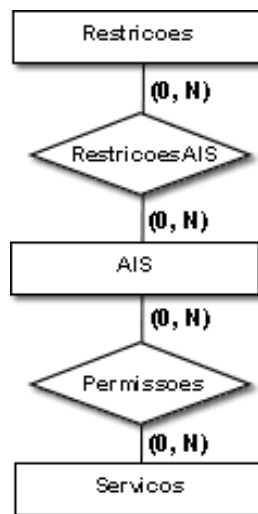
ativada ou não. Em caso negativo o acesso ao SEP de todos os usuários de suas empresas filiais é negado. Este é um atributo utilizado pelo modelo  $UCON_{preA1}$  (seção 7.2.1.1).

esta tabela possui informações fixas que são os sete dias da semana. Ela é utilizada para que possam ser configurados os horários de cada dia da semana em que cada filial tem direito de acessar o sistema.

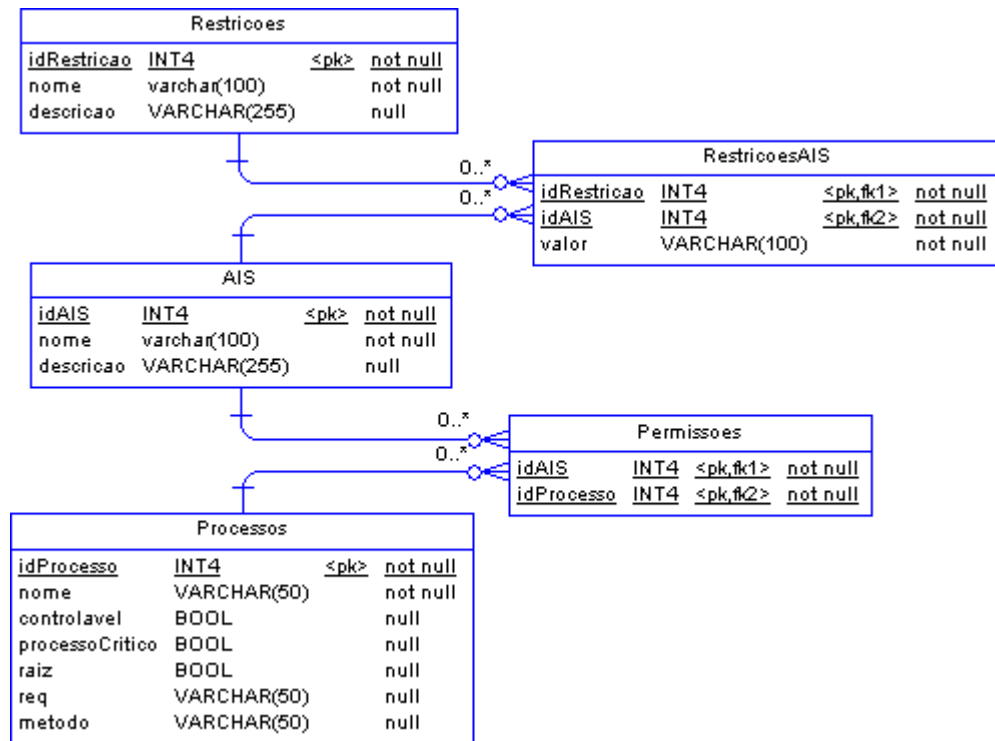
esta tabela representa um relacionamento, entre as tabelas “DiaSemana” e “Filial”, com cardinalidade de muitos-para-muitos. É através dela que são configurados os horários dos dias da semana em que os usuários de cada filial podem acessar o SEP. Desta forma, é possível configurar diferentes horários de acesso de uma filial em um mesmo dia da semana.

Através das horas de início e término, é possível verificar se a hora atual está dentro do prazo especificado. Caso a hora atual seja menor que a hora de início ou maior que a hora de término, o acesso dos usuários da filial ao SEP é negado. Estes dois horários são atributos utilizados pelos modelos  $UCON_{preC0}$  e  $UCON_{onC0}$  (seções 7.2.1.5 e 7.2.1.6).

Nestas tabelas estão as informações necessárias para a definição dos AISs. A Figura 7.5 ilustra o Diagrama de Entidade e Relacionamento – DER das tabelas que definem os AISs.



A Figura 7.6 ilustra o esquema lógico destas tabelas.



esta tabela representa o Agrupamento Implícito de Sujeitos e armazena os agrupamentos que têm acesso aos processos oferecidos pelo SEP.

esta tabela armazena as restrições exigidas por cada um dos agrupamentos cadastrados no SEP. Serão estas restrições que farão o sistema decidir a quais agrupamentos um usuário pertence e, conseqüentemente, a quais processos ele pode ter acesso.

esta tabela representa um relacionamento, entre as tabelas “Restricoes” e “AIS”, com cardinalidade de muitos-para-muitos. É através dela que são configurados os horários dos dias da semana em que os usuários de cada filial podem

acessar o SEP. Desta forma, é possível configurar diferentes horários de acesso de uma filial em um mesmo dia da semana. O atributo *valor* armazena o valor referente a cada restrição, que pode ser, por exemplo, o id de uma função, o id de uma filial ou empresa parceira e até mesmo o sexo do usuário. Ele varia de acordo com a restrição que se deseja criar. Pelo fato destes valores poderem ser de tipos variados. Optou-se por criar um campo genérico de tipo VARCHAR, de forma que o próprio sistema saberá converter os dados para seus respectivos tipos.

esta tabela armazena os processos a serem disponibilizados pelo SEP para que diferentes SEC possam acessá-los. Estes processos correspondem aos métodos de classes que foram configurados no *Web Service Apache Axis* para poderem ser acessados remotamente. O atributo de tipo booleano *controlavel* determina se deve haver ou não um controle de acesso sobre o processo. Isto porque deve haver processos que possam ser acessados por qualquer usuário, mesmo sem se autenticar. O processo de autenticação por login e senha é um exemplo, pois qualquer usuário de um SEC tem o direito de acessar este processo para poder se autenticar. Em caso negativo o acesso deve ser liberado sem que haja qualquer verificação de controle de acesso. O atributo de tipo booleano *processoCritico* determina se o processo é crítico ou não. Em caso positivo o SEP deve obrigar que o usuário forneça sua senha crítica antes que o processo seja executado. Este é um atributo utilizado pelo modelo UCON<sub>onB2</sub> (seção 7.2.1.4). O atributo de tipo booleano *processoInicio* é utilizado para determinar se o processo é o ponto de partida para outros processos ou não. Ele serve para auxiliar a formação de menus no SEC.

esta tabela representa um relacionamento, entre as tabelas “Processos” e “AIS”, com cardinalidade de muitos-para-muitos. É através dela que são configuradas as permissões de acesso que um grupo pode ter sobre os processos disponibilizados pelo SEP.

Para que seja possível implementar o modelo proposto, é de fundamental importância que se configure o Apache Axis corretamente. Com isto os processos do SEP poderão ser acessados remotamente pelo SEC e o controle de acesso poderá ser feito sobre estes processos.

Esta dissertação não possui como foco principal a interação, muito menos destacar características de utilização de *webservices*. Sendo assim, a descrição de *webservices* e sua configuração serão feitas de forma básica apenas para que seja possível compreender como o modelo proposto foi implementado.

A configuração dos processos a serem compartilhados no Axis é feita principalmente através de um arquivo no formato *Web Service Deployment Descriptor (WSDD)*. Este arquivo possui um conjunto de ferramentas necessárias para se disponibilizar um serviço, o qual pode ser constituído de vários métodos compartilhados (AXIS, 2005). Nesta dissertação estarei me referindo a estes métodos como processos. Sendo assim, um serviço pode ser constituído de um ou mais processos.

A Figura 7.7 ilustra a configuração básica de um arquivo WSDD para que seja possível compartilhar métodos de uma classe Java. As linhas 1 e 2 indicam que o arquivo XML é um descritor WSDD. A *tag* `<service>` é utilizada para configurar o serviço a ser disponibilizado. Na linha 3 é configurado o nome do serviço e o tipo de compartilhamento, que neste caso é *Remote Process Call(RPC)*. Após a inicialização da *tag*, é necessário indicar a localização da classe que irá implementar os processos a serem compartilhados. Esta indicação é feita de acordo com a linha 4. Por fim, é necessário que se configure os processos que poderão ser acessados remotamente. Esta configuração é feita como mostra a linha 5. Neste caso, todos os processos desta classe são compartilhados devido a utilização do parâmetro “\*”.

```

1 <deployment xmlns="http://xml.apache.org/axis/wsdd/"
2             xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
3   <service name="MyService" provider="java:RPC">
4     <parameter name="className" value="samples.userguide.example3.MyService"/>
5     <parameter name="allowedMethods" value="*" />
6   </service>
7 </deployment>

```

Uma vez que este arquivo tenha sido configurado, é necessário enviá-lo ao servidor Axis para que o mesmo possa reconhecer as novas configurações e atender às solicitações dos clientes. Esta ação é feita através da execução da classe "org.apache.axis.client.AdminClient". Desta forma, uma invocação típica do AdminClient se parece com:

```

% java org.apache.axis.client.AdminClient deploy.wsdd
<Admin>Done processing</Admin>

```

A mensagem “*Done processing*” aparece no *prompt* de comando indicando que os parâmetros estão corretos e que o Axis está processando o arquivo WSDD. Este comando faz com que os processos do serviço configurado possam ser acessados via SOAP.

Por este arquivo é possível gerar, através de ferramentas do Axis, o arquivo WSDL e as classes Skeleton, Stub ou Impl, que são necessários para o acesso e execução dos processos disponibilizados. Maiores informações sobre estes arquivos podem ser encontradas em (AXIS, 2005).

A configuração dos processos e do Controlador UCON<sub>ABC</sub> deve ser feita através de um arquivo WSDD. Cada módulo do sistema possui um arquivo WSDD específico. A Figura 7.8 ilustra o arquivo WSDD que configura os processos do Serviço “ServerProdutosService” que corresponde ao módulo “Produtos”.



```

1 <deployment
2   xmlns="http://xml.apache.org/axis/wsdd/"
3   xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
4
5   <service name="ServerProdutosService" provider="java:RPC" >
6
7     <requestFlow>
8       <handler type="java:org.apache.axis.handlers.JAXRPCHandler">
9         <parameter name="className"
10          value="com.server.produtos.service.handlers.UCONHandler"/>
11       </handler>
12     </requestFlow>
13
14     <parameter name="className"
15      value="com.server.produtos.service.BloqueadorProdutos"/>
16
17     <parameter name="allowedMethods" value="*/>
18
19     <beanMapping qname="ns:ProdutosModel" xmlns:ns="urn:ProdutosFacadeLocal"
20      languageSpecificType="java:com.server.produtos.model.ProdutosModel"/>
21
22   </service>
23 </deployment>

```

O código da Figura 7.8 diferencia-se em alguns aspectos do código exibido pela Figura 7.7. A configuração do tipo de serviço, da classe e dos processos a serem acessados continua a mesma coisa. As diferenças estão nas *tags* `<requestFlow>` e `<beanMapping>`.

Na linha 7 a *tag* `<requestFlow>` configura o Controlador UCON<sub>ABC</sub>. Este controlador nada mais é do que uma classe *Handler*. Este tipo de classe é utilizado pelo Axis para interceptar a solicitação de uso dos processos disponibilizados. Com isto, é possível implementar métodos de controle de acesso ou métodos de *log* que se beneficiem deste recurso. A configuração feita entre as linhas 7 e 12 permitem ao Controlador UCON<sub>ABC</sub> interceptar todas as solicitações feitas aos processos dos serviços e realizar a verificação de controle de acesso.

A configuração incorreta deste arquivo poderá acarretar falhas no sistema de controle de acesso, pois os processos poderiam estar sendo disponibilizados sem qualquer controle. Assim, todos os arquivos WSDD de todos módulos devem conter a configuração do Controlador UCON<sub>ABC</sub> através da *tag* `<requestFlow>`.

A *tag* <beanMapping> é utilizada para configurar um *Value Object* (VO) que deve ser enviado do SEP aos SECs. Um VO pode representar uma tabela de produtos e ser utilizado para armazenar seus atributos para serem enviados a um SEC em uma eventual consulta. Todo objeto enviado entre os sistemas que se interagem deve ser serializado na origem e deserializado no destino. Para que isto seja possível é necessário declarar a *tag* <beanMapping> para indicar qual objeto esta sendo enviado.

Todos os processos que forem configurados nos arquivos WSDD devem ser cadastrados no Banco de Dados na tabela “Processos” da Figura 7.5. Assim é possível relacionar os processos configurados e cadastrados aos grupos que podem acessá-los.

De acordo com a API do Apache Axis, toda classe *Handler* deve implementar a interface *Handler* e, conseqüentemente, implementar alguns métodos obrigatórios. Um destes métodos é muito importante para implementar o controle de acesso aos processos disponibilizados. Este método tem a declaração:

```
public boolean handleRequest(MessageContext context)
```

Este método é acionado pelo Axis sempre que for solicitada a execução de algum processo. Com isto, é neste método que o Controlador UCON<sub>ABC</sub> realiza suas verificações nos filtros de Autorização, Obrigação e Condição para decidir se concede o acesso ao processo ou não.

Como é possível observar, este método retorna um valor *booleano*. A API do Apache Axis afirma que se o retorno deste método for “*true*” o acesso ao processo solicitado é liberado, caso contrário, o acesso é negado.

Esta característica seria muito útil para o Controlador UCON<sub>ABC</sub>, pois bastaria retornar “*false*” que o próprio Axis se encarregaria de impedir a execução deste processo. No entanto, o Axis possui uma falha nesta característica, pois mesmo retornando um “*false*” ele permite a execução do processo.

No arquivo WSDD é possível configurar o compartilhamento de métodos de classes EJB, o que seria o ideal para a estrutura exibida pela Figura 7.2. No entanto, esta falha do Axis obriga a utilização de uma improvisação. Optou-se por um Bloqueador.

O Bloqueador nada mais é do que uma classe que acessa os métodos das classes EJB que devam ser compartilhados. Cada método do Bloqueador corresponde a um método da classe EJB. No arquivo WSDD da Figura 7.8, é possível observar que a classe configurada para ser acessada não é uma classe EJB, mas sim uma classe Bloqueadora.

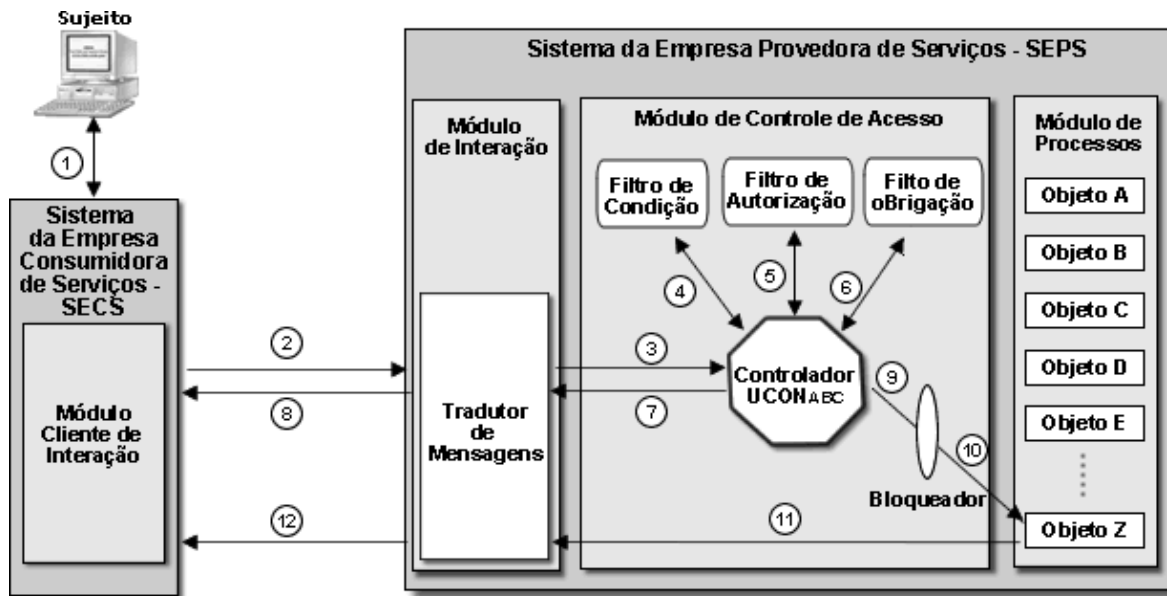
```

public ArrayList getAllProdutos() {
    ArrayList produtos = new ArrayList();
    if (acessoLiberado()) {
        try {
            ProdutosFacadeLocal produtosFacade =
                EJBGetterLocal.getProdutosFacadeLocalHome().create();
            produtos = produtosFacade.getAllProdutos();
        } catch (Exception e) {
            System.out.println(
                "error at getAllProdutos: " + e.getMessage());
            return null;
        }
        return produtos;
    } else
        return null;
}

```

A Figura 7.9 ilustra um caso em que um método “getAllProdutos()” de um Bloqueador instancia o objeto da classe “ProdutosFacadeLocal” e acessa o método “getAllProdutos()” de uma classe EJB apenas se o método “acessoLiberado()” retornar um “true”. Este último método faz a verificação de acordo com a decisão tomada pelo Controlador UCON<sub>ABC</sub>.

Com isto, o Controlador UCON<sub>ABC</sub> continua sendo um *Handler*. No entanto, ao invés dele próprio negar o acesso a um processo, ele apenas informa o Bloqueador se o acesso deve ou não ser permitido. Este processo corresponde aos passos 9 e 10 da Figura 7.10. Esta figura é praticamente a mesma que a da proposta de aplicação no Capítulo 6. A única diferença está na presença do Bloqueador.



A utilização de um Bloqueador aumenta o trabalho de programação, pois deve ser implementado um método correspondente a cada método EJB que deva ser compartilhado. Este é o “preço” que se deve pagar por esta falha do Apache Axis. Caso em futuras versões essa falha seja solucionada, a utilização de um bloqueador pode ser extinta. É por este motivo que no modelo de implementação proposto no Capítulo 6 não há a presença de um Bloqueador.

Para ser autorizado a acessar o sistema, o usuário deve passar primeiro por um processo de autenticação. Este processo é feito da forma tradicional, através do SEC, com o fornecimento do *login* e a senha do usuário.

Para que a autorização de acesso ao sistema seja concedida, são considerados os seguintes atributos do usuário:

- é verificado o campo “contaAtiva” da tabela “ContaUsuario” para verificar se sua conta esta ativada ou desativada.

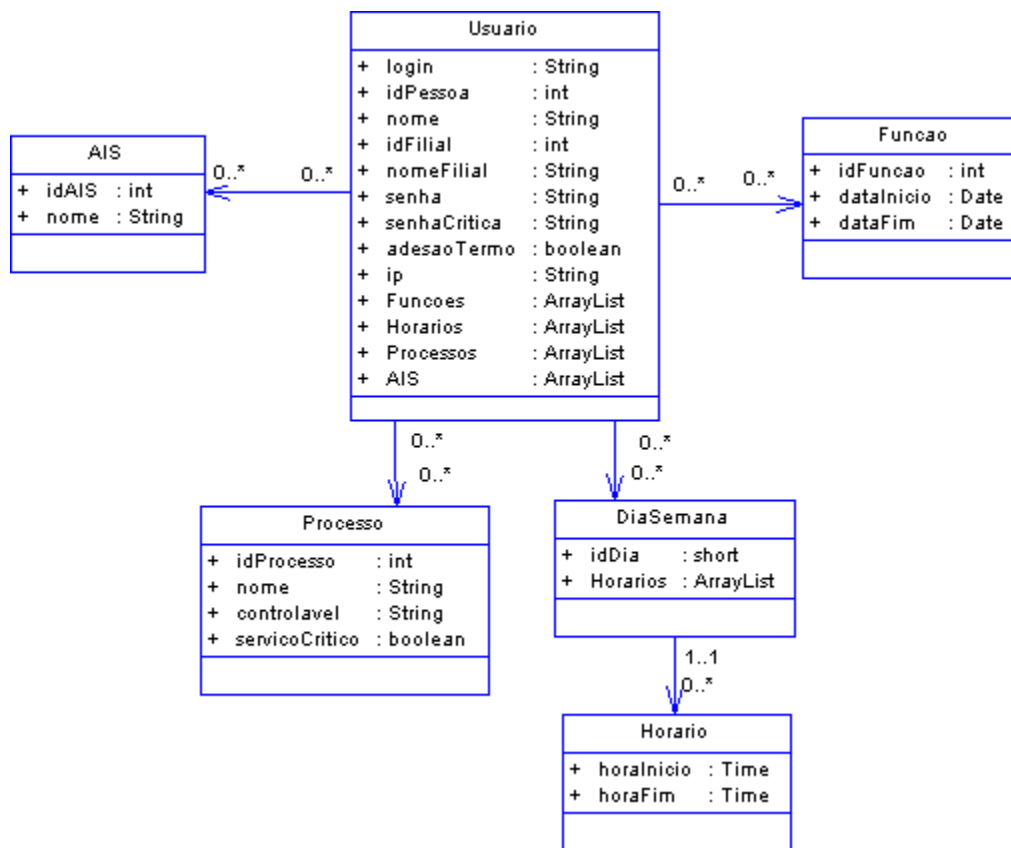
- é verificado o campo “adesaoTermo” da tabela “ContaUsuario” para verificar se o usuário já leu e aceitou o termo de adesão ao sistema. Este atributo possibilita a utilização do modelo  $UCON_{preB1}$  (seção 7.2.1.3).
- é verificado o campo “ativa” da tabela “Filial” para verificar se a filial a que o usuário pertence está ativa no SEP.
- é verificado o campo “ativa” da tabela “EmpresaParceira” para verificar se a empresa parceira a que a filial do usuário pertence está ativa no SEP.
- é verificado se o campo “ip” da tabela “EmpresaParceira” para verificar se o IP de origem da solicitação é o mesmo que o cadastrado no SEP. Este atributo possibilita a utilização do modelo  $UCON_{preC0}$  (seção 7.2.1.5).
- é verificado os campos “horaInicio” e “horaFim” da tabela “HorarioAcesso” para verificar se o horário em que o usuário está se autenticando está autorizado no sistema para a sua empresa filial. Este atributo possibilita a utilização do modelo  $UCON_{preC0}$  (seção 7.2.1.5).

Se na autenticação a senha fornecida estiver incorreta, mas é possível identificar o usuário através da combinação de seu login e IP de origem, então é possível incrementar um atributo contador para este usuário. Caso este usuário erre mais do que três vezes o fornecimento de sua senha, o SEP entende que esta é uma tentativa de ataque e então desativa a conta deste usuário. Este processo de autenticação corresponde ao modelo  $UCON_{preA1}$  (seção 7.2.1.1).

Caso a autenticação seja feita com sucesso e o acesso ao sistema seja autorizado, inicia-se então o processo de geração do objeto de sessão.

A sessão neste caso refere-se àquela criada em *Containers* de sistemas Web. Linguagens como PHP, ASP ou JSP oferecem este recurso. Através dele é possível armazenar informações de um usuário enquanto ele navega pelo sistema.

Este objeto contém todas as informações necessárias para o controle de acesso do usuário dentro do SEP. Para que o objeto de Sessão seja gerado, são necessários seis tipos de objetos diferentes que, encapsulados, formam um único objeto. Este esquema pode ser visualizado através da Figura 7.11.



As seis classes que implementam estes objetos são:

- o objeto gerado por esta classe recebe os principais dados do usuário para sua identificação e controle de acesso. Ele também possui *arrays* para armazenar os objetos das funções que possui os horários de acesso, processos que podem ser acessados e os grupos a que pertence. Este objeto, após encapsulados os demais objetos, é atribuído à sessão.
- o objeto gerado por esta classe fica armazenado em uma *array* de Funções e possui atributos que identificam a função e definem seu prazo de validade.
- o objeto gerado por esta classe possui um atributo para receber um dia da semana e uma *array* para receber os horários que o usuário pode acessar o SEP.
- o objeto gerado por esta classe possui atributos que recebem os horários de início e de término de acesso ao sistema em cada dia da semana.

- o objeto gerado por esta classe possui atributos que recebem os dados que identificam cada processo a que o usuário pode ter acesso no SEP.
- o objeto gerado por esta classe possui atributos que recebem os dados que identificam cada grupo a que o usuário pertence.

Os filtros são classes Java que são acessadas pelo Controlador  $UCON_{ABC}$  no intuito de prover o controle de acesso aos processos compartilhados de acordo com as características do modelo  $UCON_{ABC}$ . Eles são utilizados após o usuário ter se autenticado no sistema. Como foi descrito na seção 6.2, existem três tipos distintos de Filtros: Filtro de Autorização, Filtro de Obrigação e Filtro de Condição.

Este é o primeiro filtro a ser acionado pelo Controlador  $UCON_{ABC}$ . Ele realiza uma série de verificações de autorização. Caso não encontre irregularidades ele retorna o valor *booleano* “true” e o Controlador  $UCON_{ABC}$  prossegue no processo de verificação nos outros filtros, senão o processo de verificação é cancelado e o acesso ao processo é negado ao usuário. Este filtro corresponde ao modelo  $UCON_{onA2}$  (seção 7.2.1.2) e realiza três verificações de autorização: Sessão Expirada, Autorização de Acesso ao Processo e Número de Negações de Acesso.

Este filtro verifica se a sessão do usuário solicitante ainda está ativada. Em caso positivo, esta verificação autoriza a execução da próxima verificação, senão o SEP retorna ao SEC uma mensagem de “Sessão Expirada”. Com isto, o usuário deve autenticar-se novamente no SEP para poder voltar a ter acesso aos processos.

Ao autenticar-se no SEP, o usuário pode ficar inativo em um tempo máximo de 30 minutos. Ultrapassando este tempo, o próprio *Container* (Tomcat) se encarrega de desativar a sessão deste usuário.

A verificação para a autorização de acesso a um processo é feita apenas se o processo em questão possui o campo *booleano* “controlavel” da tabela “Processos” com o valor *true*.

O objeto de sessão pode ser acessado a qualquer momento por qualquer um dos três filtros. Este objeto possui um *ArrayList* contendo os processos que o usuário pode solicitar. Desta forma, todas as vezes que um processo for solicitado por um usuário, este filtro verifica se o processo solicitado encontra-se dentro desta lista de processos autorizados. Em caso positivo este filtro autoriza o acesso, senão o acesso é negado.

Todas as vezes que este filtro nega o acesso a um processo por ele não estar presente na lista de processos autorizados, um contador é incrementado e colocado na sessão do usuário. Caso este contador chegue a um número máximo de três tentativas negadas de acesso, o filtro automaticamente entende que o usuário está tentando burlar o controle de acesso de alguma forma. Com isto, ele toma medidas preventivas para evitar este tipo de ação. Estas medidas são:

- Desativar a sessão atual do usuário, evitando que o mesmo continue utilizando o sistema no momento da infração.
- Desativar a conta do usuário através do atributo “contaAtiva” da tabela *ContaUsuario* para que o mesmo não acesse mais o SEP até que seu acesso seja liberado.
- Enviar um e-mail para o usuário infrator informando-lhe o por quê da desativação da conta;
- Enviar e-mail para o usuário que tenha a função de gestor de contas da empresa que pertence o usuário infrator.

O contador de tentativas de acesso é um atributo mutável do sujeito. Este atributo não é armazenado no Banco de Dados. Ele apenas é criado quando há a irregularidade no pedido de acesso de um processo e é finalizado quando o usuário sai do SEP.



Este é o segundo filtro a ser acionado pelo Controlador UCON<sub>ABC</sub>. Ele realiza verificações de obrigação. Caso a obrigação tenha sido atendida ele retorna o valor *booleano* “true” e o Controlador UCON<sub>ABC</sub> prossegue no processo de verificação no terceiro e último filtro. Este filtro corresponde ao modelo UCON<sub>onB2</sub> (seção 7.2.1.4) e realiza duas verificações de obrigação: Processo Crítico e Excesso de Erro de senha crítica

Para realizar a verificação do processo crítico o sistema verifica o atributo do tipo *booleano* “processoCritico” da tabela “Processos”. Em caso positivo, o Controlador UCON<sub>ABC</sub> nega o acesso ao processo solicitado e o SEP envia uma mensagem ao SEC solicitando que o usuário forneça a sua senha crítica. Após o usuário enviar sua senha crítica, este filtro verifica se a senha é válida comparando-a com a do objeto de sessão. Em caso positivo, o acesso ao processo solicitado é liberado.

Caso a senha crítica fornecida pelo usuário esteja incorreta, o sistema incrementa um contador de “Erro de Senha Crítica”. Assim, inicia-se um processo similar ao do Filtro de Autorização. Caso este contador chegue a um número máximo de três tentativas erradas no fornecimento de senha crítica, o filtro automaticamente entende que o usuário está tentando adivinhar qual é a senha crítica. Com isto, ele toma as mesmas medidas preventivas do Filtro de Autorização.

Este é o terceiro e último filtro a ser acionado pelo Controlador UCON<sub>ABC</sub>. Ele realiza verificações de condição. Caso a condição tenha sido atendida ele retorna o valor *booleano* “true” e o Controlador UCON<sub>ABC</sub> finalmente autoriza o acesso ao serviço solicitado pelo usuário. Este filtro corresponde ao modelo UCON<sub>onCo</sub> (seção 7.2.1.6) e realiza três verificações de condição: Localização, Horário e Data.

é verificado se o endereço IP de origem do cliente é o mesmo que o do SEC da empresa parceira. O endereço IP do SEC deve ser real e fixo de forma que o

mesmo fique cadastrado no SEP. A verificação do IP é feita através do objeto de sessão que contém o IP que está cadastrado no banco de dados. Caso os endereços IP não sejam iguais, quer dizer que o endereço IP do banco de dados não é o mesmo do SEC que o usuário está acessando. Isto pode caracterizar uma tentativa de ataque. Sendo assim, este filtro nega o acesso deste usuário ao processo solicitado.

é verificado se o horário em que o usuário esteja tentando acessar um processo do SEP está de acordo com o que foi previamente configurado para a filial que pertence. As tabelas “DiaSemana” e “HorárioAcesso” possibilitam esta configuração. As informações dos horários permitidos são acessadas através do objeto de sessão do usuário. Caso o horário em que o usuário está solicitando a execução de um processo esteja fora do que lhe é permitido, o SEP automaticamente invalida a sessão do usuário e nega a sua autenticação.

é verificado se a data em que o usuário está solicitando a execução de um processo está entre as datas de início e término da função exercida pelo usuário no SEP. Caso a data esteja fora das datas configuradas, o usuário não mais exerce a função em questão. Como consequência, este atributo pode fazer com que o usuário não mais pertença ao grupo que pertencia anteriormente. Sendo assim, ele perde o acesso aos processos relacionados ao seu antigo grupo. Neste caso, o objeto de sessão é atualizado já com a listagem nova dos processos que o usuário pode executar.

Esta seção exibe algumas páginas do sistema de forma que sua visualização fique mais clara. Estas páginas correspondem ao SEC. Isto porque o SEP tem por obrigação, neste protótipo, apenas disponibilizar seus processos e realizar o controle de acesso sobre eles.

A Figura 7.12 ilustra a página do SEC em que o usuário deve fornecer seu login e senha para autenticar-se no SEP. Neste caso, o usuário já está autenticado no SEC.

**SEC**

Menu

- Página Inicial
- Acesso ao SEP
- Gestão de Recursos Humanos
- Gestão de Estoque
- Gestão de Contratos
- Gestão de Qualidade
- Gestão de Clientes
- Gestão de Projetos
- Gestão de Planejamento Estratégico
- Gestão de Marketing
- Sair

**SEC**

Por favor, insira seu login e senha para entrar no sistema

login:

Senha:

Entrar

A Figura 7.13 ilustra a página do termo de adesão que é apresentado ao usuário quando realiza sua primeira autenticação. Neste estágio, o acesso ao SEP ainda não foi liberado.

**SEC**

Menu

- Página Inicial
- Acesso ao SEP
- Gestão de Recursos Humanos
- Gestão de Estoque
- Gestão de Contratos
- Gestão de Qualidade
- Gestão de Clientes
- Gestão de Projetos
- Gestão de Planejamento Estratégico
- Gestão de Marketing
- Sair

**Termo de Adesão ao SEP**

Este é o segundo filtro a ser acionado pelo Controlador UCONABC. Ele realiza verificações de obrigação. Caso a obrigação tenha sido atendida ele retorna o valor booleano "true" e o Controlador UCONABC prossegue no processo de verificação no terceiro e último filtro. Este filtro realiza duas verificações de obrigação: Processo Crítico e Excesso de Erro de senha crítica. Este filtro corresponde ao modelo referenciado pela Subseção 7.2.1.4.

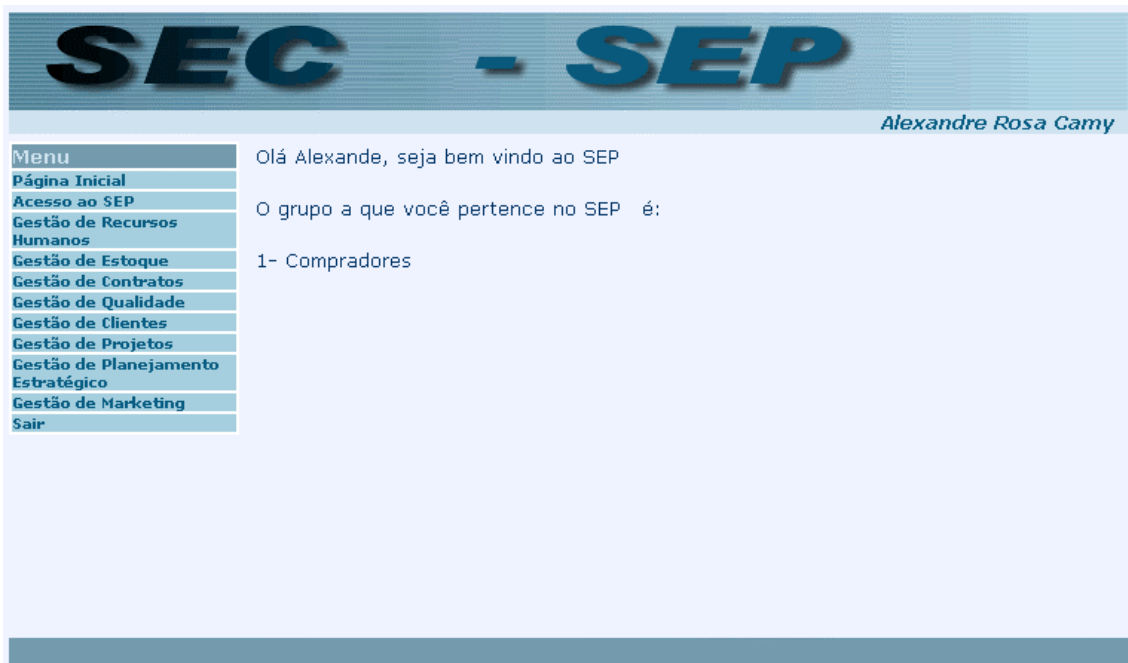
Este é o segundo filtro a ser acionado pelo Controlador UCONABC. Ele realiza verificações de obrigação. Caso a obrigação tenha sido atendida ele retorna o

Aceito Termo de Adesão

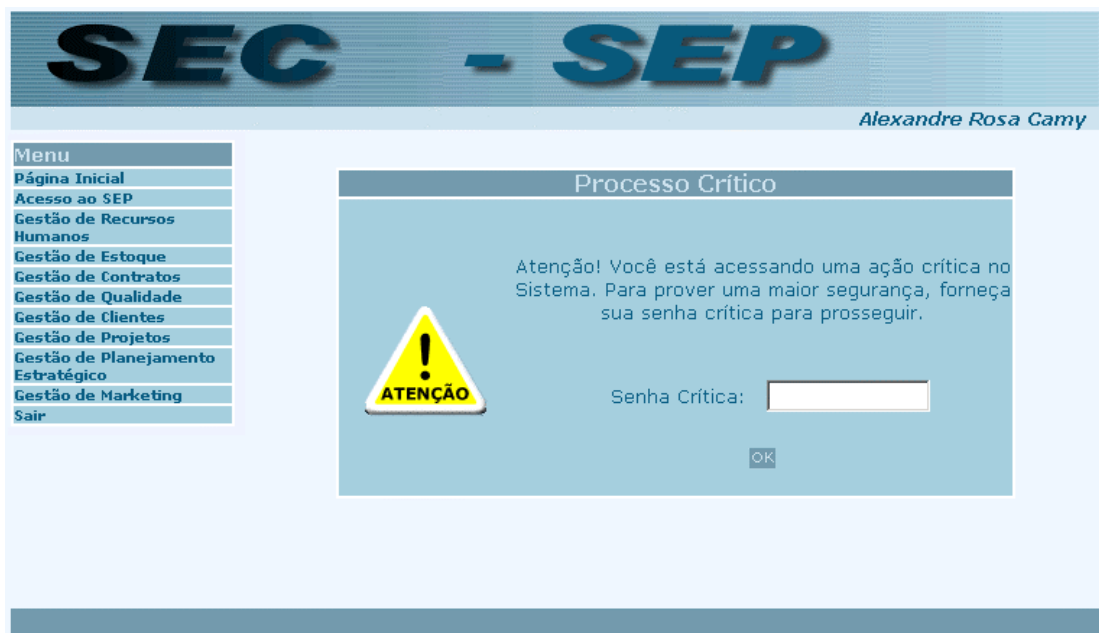
Não Aceito Termo de Adesão

OK

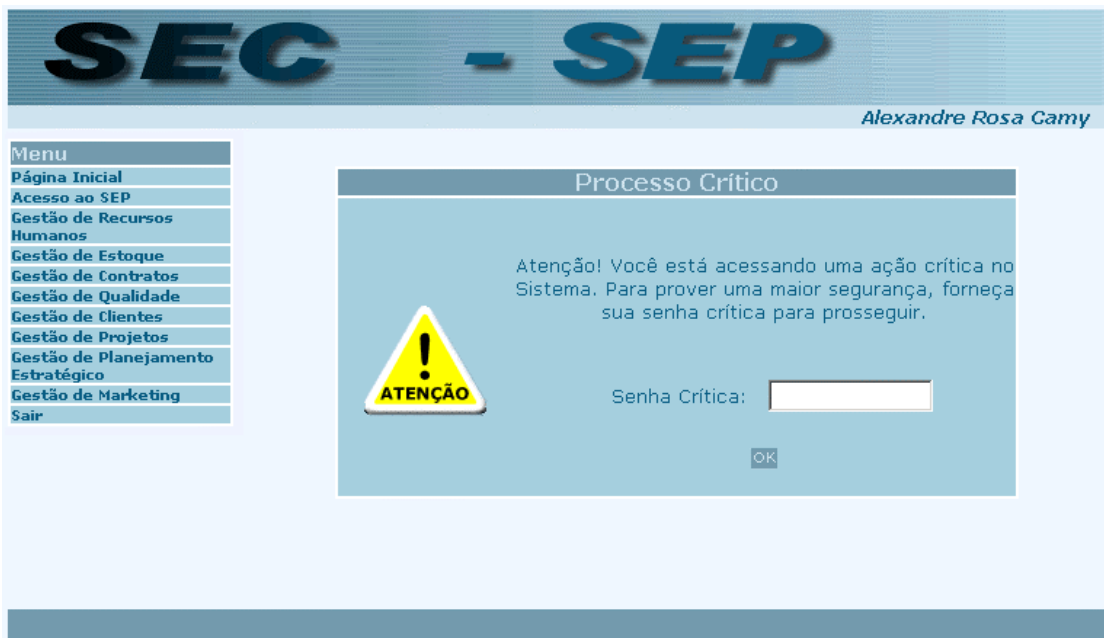
A Figura 7.14 ilustra a página inicial após o usuário ter se autenticado. Neste estágio, a sessão do usuário já foi gerada no SEP e o usuário já tem acesso aos seus processos.



A Figura 7.15 ilustra a página onde o usuário deve fornecer sua senha crítica antes de acessar um processo crítico.



A Figura 7.16 ilustra uma página que gera um relatório dos produtos oferecidos pelo SEP.



Por fim, A Figura 7.17 ilustra uma página de Acesso Não Autorizado, que é exibida quando o Controlador  $UCON_{ABC}$  não libera o acesso a um determinado processo.



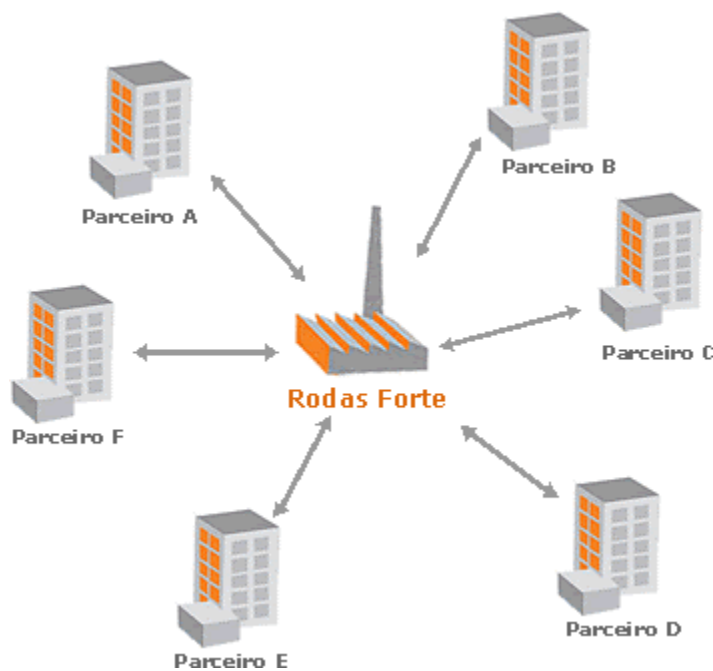
Este capítulo apresentou as tecnologias utilizadas e a descrição do desenvolvimento do protótipo do sistema de CE B2B que interagem entre si utilizando o modelo  $U\text{CON}_{ABC}$ . Observou-se que a implementação deste tipo de sistema, tendo o  $U\text{CON}_{ABC}$  como modelo de controle de acesso, é perfeitamente possível. Suas novas características contribuem para o aumento do nível de segurança neste tipo de sistema, pois o usuário é constantemente monitorado e qualquer irregularidade que ocorra ele poderá ser punido, mesmo já tendo se autenticado no sistema. Além disto, o usuário é obrigado a cumprir determinadas obrigações antes de acessar processos críticos.

Este capítulo apresenta um estudo de caso para exemplificar a aplicação da proposta desta dissertação em um ambiente de CE B2B. É importante salientar que os nomes das empresas, produtos e sistemas abordados neste estudo de caso são meramente ilustrativos e não possuem qualquer relação com o mundo real.

*Rodas Forte* é o nome de uma indústria brasileira fabricante de rodas esportivas para automóveis. Sua produção de rodas atende tanto o mercado interno quanto externo. Seus principais clientes são as indústrias de automóvel, onde os veículos fabricados já saem de fábrica com os produtos da *Rodas Forte*. Para reduzir seus custos e aumentar a eficiência nas transações entre a *Rodas Forte* e suas empresas clientes, foi desenvolvido um sistema de CE B2B, denominado RF-B2B (*Rodas Forte Business-to-Business*).

O RF-B2B segue uma tendência mundial de se estender a funcionalidade dos sistemas de gestão empresariais (ERP) tradicionais. Sendo assim, o RF-B2B não tem por objetivo substituir o sistema ERP já existente na *Rodas Forte*, mas sim agregar funcionalidades através de um trabalho em conjunto. Ao aderir a utilização de um sistema B2B, a *Rodas Forte* passou a chamar suas empresas clientes de “empresas parceiras”.

A figura 8.1 ilustra as relações existentes entre a *Rodas Forte* e suas empresas parceiras. Através dela é possível perceber que o RF-B2B é um sistema que centraliza a interação entre as empresas envolvidas. Sendo assim ele atua como um sistema servidor de serviços.



O RF-B2B possui as seguintes funcionalidades:

- o RF-B2B foi desenvolvido para prover a interação entre ele e os sistemas das empresas parceiras. Desta forma, independentemente da tecnologia utilizada, o sistema B2B de qualquer empresa parceira da *Rodas Forte* pode interagir com o RF-B2B.

- para viabilizar a negociação entre as empresas parceiras, o RF-B2B possibilita a visualização de produtos com seus preços, garantias, quantidades no estoque, dentre outros.

- o RF-B2B, disponibiliza serviços onde é possível enviar e receber ofertas para o fechamento de um negócio.

- o RF-B2B possibilita que as empresas realizem trocas de arquivos de contrato para que, após o comum acordo entre ambas, possam assinar estes contratos.



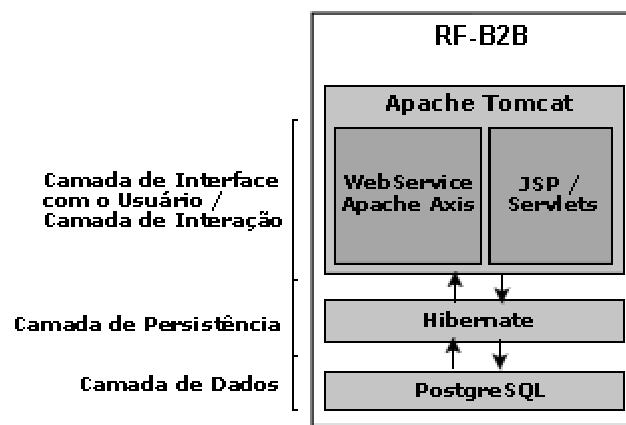
Após o estabelecimento de contrato, uma empresa parceira realizar e cancelar pedidos de entregas de produtos através do RF-B2B.

tanto os usuários da Rodas Forte, quanto os usuários das empresas parceiras podem visualizar um histórico de transações realizadas entre as empresas em um determinado período de tempo.

esta ferramenta permite o cadastro e gerência dos dados de usuários e suas respectivas funções na empresa. Tais dados são utilizados pela ferramenta de controle de acesso.

alguns grupos e suas respectivas restrições podem ser configurados através desta ferramenta.

Os desenvolvedores do RF-B2B optaram pela utilização da tecnologia Java como base para o seu desenvolvimento. Além de Java foram outras tecnologias livres. A figura 8.2 ilustra estas tecnologias:

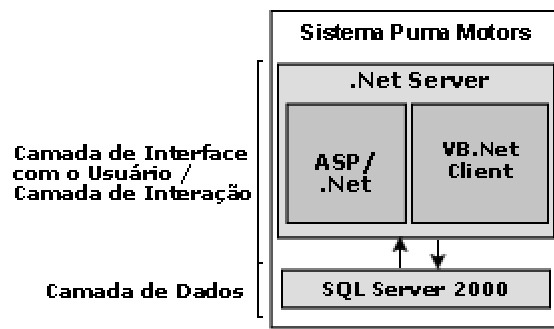


Para camada de dados foi utilizado o PostgreSQL. A camada de persistência de dados foi desenvolvida através da ferramenta Hibernate. A camada de interface com o usuário e de interação estão no mesmo nível, pois são executadas pelo servidor Apache Tomcat. Para a interface com o usuário foram utilizadas as tecnologias Java Servlets e JSP. Para a

interação entre o RF-B2B e os sistemas das empresas parceiras foi utilizado o *Webservice* Apache Axis.

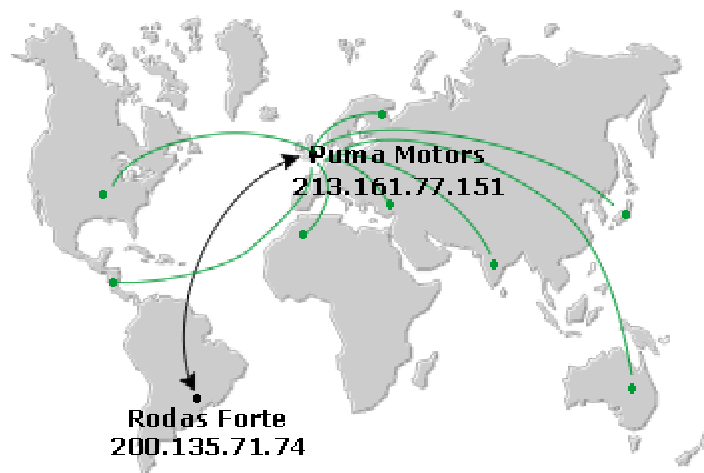
Apesar de realizar negócios com diversas indústrias automobilísticas parceiras, a *Rodas Forte* possui como principal parceira a *Puma Motors*, uma indústria Inglesa que possui filiais em diversos países do mundo.

A *Puma Motors* foi a primeira empresa parceira da *Rodas Forte* a desenvolver um módulo adicional ao seu sistema de gestão empresarial que interagisse com o RF-B2B. Seu sistema de gestão foi desenvolvido utilizando tecnologias da Microsoft. A figura 8.3 ilustra as tecnologias utilizadas.



Para a camada de dados foi utilizado o banco de dados SQL Server 2000. O Servidor .Net é utilizado para executar o sistema, que foi desenvolvido utilizando as tecnologias ASP / .NET. A tecnologia .NET possui ferramentas que possibilitam o sistema comunicar-se com outros através de *Webservices*, ou seja, ela possui meios de tornar-se um cliente *Webservice*. Desta forma, mesmo a tecnologia utilizada pela *Puma Motors* sendo completamente diferente da utilizada pela *Rodas Forte*, os sistemas das duas empresas podem interagir entre si através da utilização de um *Webservice*.

Apesar de ter sede na Inglaterra, a *Puma Motors* possui fábricas filiais espalhadas por todo o mundo. Cada uma destas filiais tem acesso ao sistema RF-B2B através do sistema central da *Puma Motors*. A figura 8.4 ilustra a localização geográfica das empresas.



A *Rodas Forte* está situada no Brasil e o sistema RF-B2B atende pelo IP 200.135.71.74. Assim, os sistemas das empresas parceiras podem interagir com o RF-B2B através deste IP. A *Puma Motors*, com sede na Inglaterra, possui seu sistema central que atende pelo IP 213.161.77.151.

Os sistema central da *Puma Motors* possui uma *extranet* onde todas as filiais podem trocar informações com a matriz. Assim, a linha preta da figura 7.21 representa a interação entre os sistemas da *Rodas Forte* e a *Puma Motors*. A linha verde representa a troca de informações entre a *Puma Motors* e suas filiais.

Desta forma, qualquer uma das filiais tem autonomia para realizar negociações com a Rodas Forte. Este cenário caracteriza um ambiente de CE B2B de sistemas que interagem entre si.

Após realizar uma pesquisa, a equipe de desenvolvimento do RF-B2B definiu que o UCON<sub>ABC</sub> deveria ser aplicado como modelo no desenvolvimento do controle de acesso do sistema. Além disso, a equipe constatou que a proposta desta dissertação enquadrava-se nas particularidades de controle de acesso necessárias ao RF-B2B.

O gerenciamento de permissões do RF-B2B leva em consideração a empresa parceira, filial da empresa parceira e a função do usuário na empresa. Desta forma, o sistema de controle de acesso do RF-B2B pode atribuir diferentes permissões entre: i) diferentes empresas; ii) diferentes filiais de uma mesma empresa e iii) diferentes funções dentro de uma mesma filial. Nestes itens são considerados os atributos do usuário e sua combinação pode determinar a participação de um usuário em um determinado grupo de acesso do sistema. A tabela 8.1 ilustra alguns dos grupos existentes e suas respectivas restrições configuradas no sistema de controle de acesso do RF-B2B.

Grupo	Restrições			Serviços
	Empresa	Filial	Função	
RF1				Gestão de dados pessoais
RF2			Gestor de Permissões	Gestão de permissões em seu ambiente
RF3			Gestor de RH	Gestão de dados profissionais dos usuários
RF4	Puma Motors		Negociador	Gestão de dados de negociação da Puma Motors
RF5	Puma Motors		Comprador	Gestão de pedidos de compra da Puma Motors
RF6	Puma Motors		Gerente de Compras	Gestão de histórico de compras da Puma Motors
RF7	Puma Motors	Puma - USA	Gestor de Contratos	Gestão de Contratos da Puma – USA
RF8	Puma Motors	Puma - México	Gestor de Contratos	Gestão de Contratos da Puma - México
RF9	Puma Motors	Puma - Rússia	Gestor de Contratos	Gestão de Contratos da Puma - Rússia
RF10	Puma Motors	Puma - Turquia	Gestor de Contratos	Gestão de Contratos da Puma - Turquia
RF11	Puma Motors	Puma - Marrocos	Gestor de Contratos	Gestão de Contratos da Puma - Marrocos
RF12	Puma Motors	Puma - Índia	Gestor de Contratos	Gestão de Contratos da Puma – Índia
RF13	Puma Motors	Puma - Japão	Gestor de Contratos	Gestão de Contratos da Puma - Japão
RF14	Puma Motors	Puma - Austrália	Gestor de Contratos	Gestão de Contratos da Puma - Austrália
RF15	Alfa Motors		Negociador	Gestão de dados de negociação da Alfa Motors
RF16	Alfa Motors		Revisor de Contratos	Gestão de Contratos da Alfa Motors
...	...	...	...	...

Através da tabela 8.1 é possível observar que não há qualquer restrição para que o usuário pertença ao grupo RF1. Assim, todos os usuários que acessam o RF-B2B têm acesso às funcionalidades que este grupo tem permissão, independentemente da empresa, filial ou função.

Os grupos RF2 e RF3 possuem apenas a restrição de que o usuário possua as funções especificadas. Com isto, todos os usuários que possuam estas funções têm acesso às funcionalidades que estes grupos têm permissão, independentemente da empresa ou filial.

Os grupos RF4, RF5 e RF6 não possuem restrição com relação à filial que um usuário pertence. Assim, para o usuário pertencer a um destes grupos é necessário que ele possua uma das respectivas funções e que seja funcionário da *Puma Motors*.

Na *Puma Motors* houve a necessidade de os usuários com função de “Gestor de Contratos” tivessem diferentes permissões que variavam de acordo com as filiais a que pertenciam. Isto porque as leis de contratos variam em cada país. Por isto a necessidade de se criar os grupos cujas restrições levassem em consideração estes três fatores.

Os grupos posteriores já pertencem a outras empresas parceiras da *Rodas Forte*. Esta tabela na realidade possui um tamanho muito superior ao apresentado. Seus dados foram resumidos de forma a apenas ilustrar seu funcionamento.

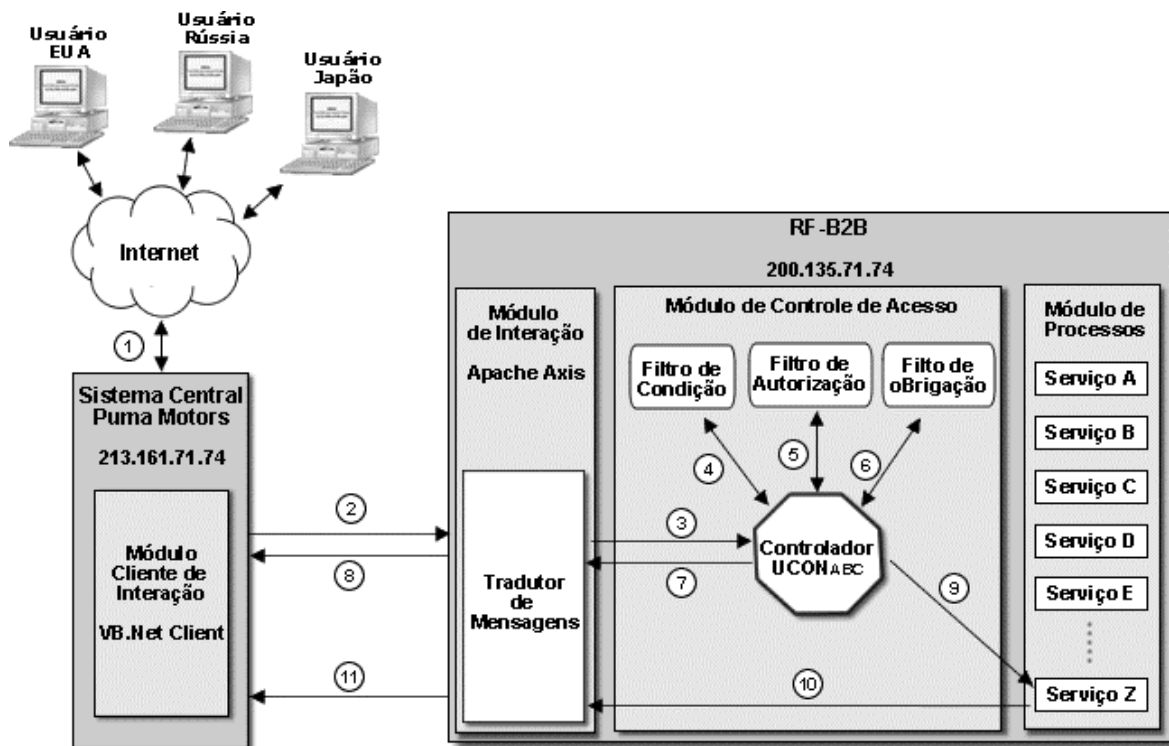
Caso um usuário da *Puma Motors* seja transferido, por exemplo, da filial do México para a filial da Turquia, seu grupo será automaticamente modificado do RF8 para o RF10.

O gerenciamento de permissões é fundamental para o perfeito funcionamento do sistema de controle de acesso do RF-B2B.

O usuário José Silva trabalha atualmente na filial do Japão da *Puma Motors*. Para que José Silva possa trocar informações com a *Rodas Forte* ele deve, primeiramente, estar autenticado na *extranet* do sistema central da *Puma Motors*. Posteriormente ele deve autenticar-se novamente, mas agora no RF-B2B, para então ser possível a troca de informações.

No processo de autenticação, é verificado primeiramente se o IP de origem do pedido de autenticação é o mesmo que está relacionado à *Puma Motors*, cadastrado no RF-B2B. Este é o processo de autenticação do sistema cliente adotado pelo RF-B2B. Em caso positivo é liberado o processo de autenticação de José Silva. Quando o processo de autenticação do sistema central da *Puma Motors* é finalizado sem erros, o sistema de controle de acesso do RF-B2B inicia então o processo de busca das permissões as quais José Silva tem acesso.

Primeiramente é verificado, baseando-se nos atributos de José Silva, a quais grupos do AIP ele pertence. Definido os grupos, é então feita uma busca pelas permissões de acesso as quais estes grupos têm acesso. Estas permissões são alocadas na memória do sistema, para posteriormente serem acessadas pelo Filtro de Autorização e Obrigação enquanto José Silva estiver utilizando o sistema.



De acordo com a figura 8.5, quando, por exemplo, o José Silva deseja realizar um pedido de produtos, ele envia uma solicitação via HTTP para a *extranet* da *Puma Motors* (passo 1). O Módulo Cliente de Interação recebe esta solicitação e, através da ferramenta VB.Net Client, se encarrega de encaminhá-la para o RF-B2B (passo 2).

O Módulo de Interação do RF-B2B recebe esta solicitação pelo *Webservice Apache Axis* e a traduz para a tecnologia Java. A partir deste ponto, a solicitação é enviada para o Controlador UCON<sub>ABC</sub> (passo 3), que verifica as regras de condição, autorização e obrigação do RF-B2B através de seus respectivos filtros (passos 4, 5 e 6). O filtro de condição analisa o endereço IP 213.161.77.151, que é proveniente do sistema da *Puma Motors*. Além disto, este filtro analisa os horários a que José Silva têm direito de acesso no RF-B2B. O filtro de autorização analisa as permissões que correspondem ao grupo a que o José Silva pertence. O filtro de obrigação analisa a necessidade do fornecimento de uma senha crítica para a execução de um determinado serviço.

Em caso de negação de acesso, o Controlador UCON<sub>ABC</sub> retorna ao Módulo de Interação uma resposta de negação de acesso ou de necessidade de cumprimento de alguma obrigação (passo 7). No Módulo de Interação esta resposta é traduzida em uma mensagem que é retornada ao sistema da *Puma Motors* (passo 8).

Caso a solicitação passe pela verificação de todos os filtros, então o acesso ao serviço de pedidos é liberado (passo 9). Após ser acessado e executado, o serviço de pedidos de produtos retorna ao Módulo de Interação um valor lógico do tipo verdadeiro/falso informando do sucesso ou fracasso da operação (passo 10). Após converter o valor de retorno em uma mensagem, o Módulo de Interação do RF-B2B a envia para o sistema da *Puma Motors* (passo 11) que, por sua vez traduz novamente o conteúdo para a tecnologia ASP/.Net e exibe a resposta para José Silva através de sua *extranet*.

Este capítulo apresentou um estudo de caso onde foi possível visualizar, através de um exemplo do mundo real, a aplicação do modelo UCON<sub>ABC</sub> em sistemas de CE B2B. Com isto, foi possível perceber a viabilidade de aplicação do UCON<sub>ABC</sub> neste tipo de sistema.

Diversas estratégias para controle de acesso tem sido propostas há algumas décadas e vêm crescendo e melhorando de acordo com as necessidades exigidas pelas inovações tecnológicas. O  $U\text{CON}_{ABC}$  é um modelo promissor por lidar com aspectos do controle de acesso até então ignorados pelos modelos tradicionais. No entanto, pelo fato de sua apresentação para a comunidade científica ser recente, não existem estudos práticos sobre a aplicabilidade de seus conceitos em sistemas do mundo real.

Os trabalhos relacionados se restringem a definir um esquema, *framework* ou técnica de controle de acesso para sistemas de CE B2B, ou então vincular um modelo de controle de acesso a uma ferramenta específica. Este trabalho diferencia-se dos demais por definir, como principal contribuição científica, uma forma de aplicação do modelo  $U\text{CON}_{ABC}$  em sistemas de CE B2B que interagem entre si. Com isso, tornou-se mais fácil a visualização de como é possível aplicar o  $U\text{CON}_{ABC}$  em sistemas onde é necessária uma interação, independentemente da tecnologia de interação utilizada.

A necessidade de se definir maneiras eficientes de implementar o controle de acesso neste tipo de sistema é justificada pelo fato da interação possibilitar que usuários externos acessem informações de uma empresa pelo intermédio de outros sistemas.

Foi atribuída ao modelo de implementação proposto uma forma de gerenciamento de permissões. Modelos como MAC, DAC ou RBAC são possibilidades que poderiam ser utilizadas para este gerenciamento. O modelo RBAC seria uma boa alternativa se não fossem as limitações destacadas por (ROBISON, 2002) e (GOODWIN, GOH & WU, 2002). O Agrupamento Implícito proposto por (GOODWIN, GOH & WU, 2002) é uma forma interessante de gerenciamento de permissão. No entanto, observou-se que o agrupamento de objetos desta técnica gerava uma sobrecarga administrativa desnecessária. Sendo assim, foi sugerida a criação do Agrupamento Implícito Parcial como gerenciamento



de permissões na proposta de aplicação do  $UCON_{ABC}$  em sistemas B2B. A pesquisa e análise de modelos, a explicação do Agrupamento Implícito utilizando exemplos e formas diferentes de ilustração, o levantamento de desvantagens do Agrupamento Implícito e a sugestão do Agrupamento Implícito Parcial são também contribuições científicas desta dissertação.

Foi apresentado o conceito de “sujeito composto” para modelos de controle de acesso. Este conceito surgiu de particularidades existentes entre sistemas que interagem entre si. Nestes tipos de sistema o controle de acesso deve analisar tanto os atributos de um usuário, quanto de seu sistema cliente para então decidir se concede ou não o acesso a um determinado objeto.

Observou-se durante o processo de pesquisa que é importante analisar os diferentes pontos de vista que se pode ter de um objeto quando o sistema em questão é um CE B2B. Sendo assim, esta pesquisa realizou uma análise que possibilita identificar quais modelos  $UCON_{ABC}$  melhor se adaptam no desenvolvimento de sistemas CE B2B. Os modelos utilizados nesta dissertação foram:  $UCON_{preA1}$ ,  $UCON_{onA2}$ ,  $UCON_{preB1}$ ,  $UCON_{onB2}$ ,  $UCON_{preC0}$  e  $UCON_{onC0}$ .

A utilização de filtros para executar as verificações de autorização, obrigação e condição é outra contribuição científica deste trabalho no sentido de se estruturar e organizar as classes que realizam estas verificações.

As principais decisões de projeto de implementação foram as escolhas pelas tecnologias Java e *Web service* Axis. A tecnologia Java foi escolhida por possuir uma série de ferramentas, dentro da arquitetura J2EE, que possibilitam implementar sistemas robustos com suporte a transação de processos, segurança ou balanceamento de carga. O *Web service* Axis foi escolhido como meio de interação por possuir ferramentas que otimizam o processo de desenvolvimento da interação e, além disso, auxiliam no controle de acesso. Estas características proporcionam ao desenvolvedor uma baixa preocupação com a implementação dos mecanismos de interação e facilidades a mais no desenvolvimento do sistema de controle de acesso. Desta forma, é possível dedicar uma maior atenção no desenvolvimento das funcionalidades oferecidas pelo sistema fornecedor de processos.

Apesar do Axis possuir ferramentas que facilitam o desenvolvimento do controle de acesso, ele possui uma grava falha na classe *Handler* que obrigou a utilização de um Bloqueador. A notificação desta falha e a sugestão de uma solução também são contribuições deste trabalho.

Por fim, foi apresentado um estudo de caso entre empresas parceiras da indústria automobilística. Este estudo contribuiu para o melhor entendimento da aplicabilidade da proposta desta dissertação em sistemas de CE B2B do mundo real.

A proposta da forma de aplicação do  $UCON_{ABC}$  em sistemas de CE B2B foi a principal contribuição científica desta dissertação. Para este modelo foi feita uma pesquisa para encontrar qual a melhor forma de se gerenciar as autorizações. O Agrupamento Implícito Parcial foi escolhido como forma de gerenciar as autorizações nesta dissertação.

Para trabalhos futuros é sugerida a pesquisa na utilização do Gerenciamento de Confiança como forma de gerenciar autorizações. Em conseqüência também sugiro que seja feita uma análise comparativa da forma de aplicação do  $UCON_{ABC}$  em sistema B2B, proposta por esta dissertação, com uma outra proposta que utilize o Gerenciamento de Confiança.

AXIS. “ ”. Acessado em: 05 de Janeiro de 2005, <http://ws.apache.org/axis/java/user-guide.html>, 2005.

BLAZE, Matt; FEIGENBAUM, Joan. E LACY, Jack. “ ”, em Proceedings of the 1996 IEEE Symposium on Security and Privacy. p. 164. ISBN:0-8186-7417-2 Publisher: IEEE Computer Society Press. Washington, DC, USA, 1996.

BLAZE, Matt et All. “ ”, em Secure Internet Programming: Security Issues for Mobile and Distributed Objects. Vol. 1603, p. 185-210, ISBN:3-540-66130-1, London, UK, 1999.

BLAZE, Matt et All. . Versão 2. *RFC-2704*. IETF, Setembro de 1999.

BLODGET, Henry; MCCABE, Edward. , Acessado em 20 de Novembro de 2004, <http://www.visi.com/~keefner/pdfs/ml020700.pdf>, Merrill Lynch and Company, February de 2000.

BODOFF, Stephanie et All. . ISBN: 85-352-1077-6, p. 445, Ed. Campus, Rio de Janeiro – Brasil, 2003.

BRODIE, Michael. . Em Information System Engineering: State of the Art and Research Themes. Londres, Inglaterra, 2000.

DABOUS, Feras; RABHI, Fethi; RAY, Pradeep. . Em Annual Review of Communications. ISBN 1-931695-22-9. Vol. 56, International Engineering Consortium, 2003.

DoD - Department of Defense (USA), Department of Defense Trusted Computer System, Evaluation Criteria, DOD 5200.78 - STD, Dezembro de 1985.

- DOGAC, Asuman.; CINGIL, Ibrahim  
 . Em ACM SIGecom Exchanges. Vol.2, nº 2,  
 p.16-27, Spring, 2001.
- DUHL, Joshua; KEVORKIAN, Susan. . IDC White Paper.  
[www.intertrust.com/main/research/whitepapers/IDCUnderstandingDRMSystems.pdf](http://www.intertrust.com/main/research/whitepapers/IDCUnderstandingDRMSystems.pdf).  
 2001.
- ESSMAYR, Wolfgang; PROBST, Stefan & WEIPPL, Edgar.  
 . Electronic Commerce Research. ISSN: 1389-5753. Vol 4, nº 1-2, P. 127-  
 156. Áustria, Janeiro de 2004.
- FEGHHI, Jalal; FEGHHI, Jalil. .Obtido em:  
<http://www.informit.com/articles/article.asp?p=20999>. Março de 2001.
- FERRAIOLO, David; KUHN, Richard. . In Proc. of the NIST-  
 NSA National Computer Security Conference, p. 554-563, 1992.
- FERRAIOLO, David et. All. .  
 ACM Transactions on Information and System Security, Vol. 4, Nº 3. New York,  
 Agosto de 2001.
- GANTA, Srinivas.  
 . 139 f. Tese de Doutorado em Tecnologia da Informação - Universidade  
 George Mason. Virginia – USA, 1996.
- GOODWIN, Richard; GOH, Sweefen.; WU, Frederick.  
 . IBM Systems Journal, Vol. 41, Nº 2, p.  
 303 - 317.Janeiro de 2002.
- HARDEE, Martin et All. . Acessado em 16 de Novembro de 2004,  
<http://java.sun.com/java2/whatis/>.
- HOGH Khogg; et All.  
 , em: 27th conference on Australasian computer science, p. 331-340, Vol.  
 26, Dunedin, Nova Zelândia, 2004.

- KOENEN, Rob; LACY, Jack; MACKAY, Michael.  
 . IDC Research Paper. [http://www.intertrust.com/main/research/whitepapers/Interoperable\\_DRM.pdf](http://www.intertrust.com/main/research/whitepapers/Interoperable_DRM.pdf). 2004.
- KRAFT, Reiner.  
 . In Proceedings of the ACM Workshop on XML Security. Fairfax. ISBN:1-58113-632-3, Vol. 36 - 52 , VA, USA, Novembro de 2002.
- LANDWEHR, Carl, E. International Journal of Information Security. Vol. 1, nº 1, p. 3 – 13, ISSN: 1615-5262, Agosto de 2001.
- LANDWEHR, Carl, E. ACM Computing Surveys. Vol. 13, nº 3, p. 247-278, ISSN:0360-0300, ACM Press, New York - USA Setembro de 1981.
- LAMPSON, Butler. . 5th Princeton Symposium of Information Sciences and Systems, Universidade de Princeton, Março de 1971, p. 437—443, re-impresso em Operating System Review, Janeiro de 1974, p. 18-24.
- LIU, Qiong. et. All. , em: Australasian Information Security Workshop Conference, ACM. Vol. 21, p. 49-58, ISBN ~ ISSN:1445-1336 , 1-920682-00-7. Adelaide, Austrália, 2003.
- MEDJAHED, Brahim, et. All.  
 Em The VLDB Journal — The International Journal on Very Large Data Bases. ISSN:1066-8888. Vol. 12, p. 59-85. Springer-Verlag New York. Maio de 2003.
- NEWMAN, Alexander, et. All.  
 ISBN: 85-352-01098-3, p. 861, Ed. Campus, Rio de Janeiro – Brasil, 1997.
- O'CONNELL, Brian.  
 ISBN:85.346.1244-7. Ed. Makron Books, São Paulo, Brasil, 2002.
- PARK, Jaehong.  
 .. 155 f. Tese de Doutorado em Tecnologia da Informação - Universidade George Mason. Virginia – USA, 2003.
- PARK, Jaehong; SANDHU, Ravi. . Proceedings of 9<sup>th</sup> ACM Symposium on Access Control Models and Technologies, 2004.

PARK, Jaehong. e SADHU, Ravi.

SACMAT - Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies. ACM Press. p. 57 – 64, New York – USA, 2002.

POSTGRESQL. . Acessado em 06 de Janeiro de 2005, <http://www.postgresql.org/docs/7.3/interactive/user.html>, 2002.

QUIX, Christoph; SCHOOP, Mareike e JEUSFELD, Manfred

. SIGMOD. Vol 31, N° 1, p. 49 – 54, Março de 2002.

RADOWIISKY, Zinovy.

. Em Proceeding of the 87th Annual International Supply Management Conference, Institute of Supply Management. São Francisco – USA, 2002

ROBISON, Lyn.

. Implementing B2B Commerce with .NET: A Guide for Programmers and Technical Managers. Capítulo 7, ISBN: 0201719320, Edição 1, Editora: Addison Wesley Professional, 21 de Dezembro de 2001. Obtido em: <http://www.awprofessional.com/articles/article.asp?p=27143>. Junho de 2002.

SAMARATI, Pierangela; VIMEREATI, Sabrina.

. 7 International School on Foundations of Security Analysis and Design on Foundations of Security Analysis and Design: Tutorial Lectures. p.137-196, 01 de Setembro de 2000.

SANDHU, Ravi.

. In Computer. Vol. 29, N° 2, p. 38-47, ISSN:0018-9162 . IEEE Computer Society Press, Los Alamitos/CA – USA. Fevereiro de 1996.

SANDHU, Ravi.

. Proceedings of the International Workshop on Information Assurance in Computer Networks: Methods, Models, and Architectures for Network Security. ISBN:3-540-42103-3. Vol. 2052, p. 22 – 26, London – UK, 2001.

SANDHU, Ravi e PARK, Jaehong.

. MMMACNS: International Workshop on Methods, Models and Architectures for Network Security, LNCS, p. 17-31, St. Petersburg – Russia, 2003.

SANDHU, Ravi; SAMARATI, Pierangela.

. IEEE

Communications: Volume 32, N° 9, Setembro de 1994.

SANDHU, Ravi; SAMARATI, Pierangela.

.

ACM Computing Surveys (CSUR): Volume 28, Páginas: 241 – 243, Nova York – SA, 1996.

SCHOOP, Mareike; KOLLER, Jork.; QUIX, Christoph.

. First

IFIP Conf. On E-Commerce, E-Government, E-Business (I3E). Suíça, 2001.

TAYLOR, Luke.

. Acessado em 06 de Janeiro de 2005,

[http://docs.jboss.org/jbossas/getting\\_started/jbossj2ee.pdf](http://docs.jboss.org/jbossas/getting_started/jbossj2ee.pdf), 2004.

TOMCAT.

Acessado em 06 de Janeiro de 2005,

<http://jakarta.apache.org/tomcat/tomcat-5.0-doc>, 2003.

VIMERATI, Sabrina; PARABOSCHI, Stefano; SAMARATI, Pierangela.

. Software-Practice & Experience. Vol. 33, N° 5, p. 397-421,

ISSN:0038-0644, New York, NY, USA, Março de 2003.