UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

CONTROLE DE ACESSO A RECURSOS COMPUTACIONAIS DE FORMA FLEXÍVEL E DINÂMICA ATRAVÉS DE CONTEXTO

DISSERTAÇÃO DE MESTRADO

JUNIOR MARCOS BANDEIRA

Santa Maria, RS, Brasil
2010

CONTROLE DE ACESSO A RECURSOS COMPUTACIONAIS DE FORMA FLEXÍVEL E DINÂMICA ATRAVÉS DE CONTEXTO

por

Junior Marcos Bandeira

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Qualidade e Produtividade, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Mestre em Engenharia de Produção.

Orientador Prof. Dr. Raul Ceretta Nunes

Santa Maria, RS, Brasil

2010

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

CONTROLE DE ACESSO A RECURSOS COMPUTACIONAIS DE FORMA FLEXÍVEL DE DINÂMICA ATRAVÉS DE CONTEXTO

elaborada por
Junior Marcos Bandeira

como requisito parcial para obtenção do grau de
Mestre em Engenharia de Produção

Comissão Examinadora:

_____
Raul Ceretta Nunes, Dr. (UFSM)
(Presidente/Orientador)

_____
Fabio Luciano Verdi, Dr. (UFScar)

_____
Roseclea Medina Duarte, Dra. (UFSM)

Santa Maria, 30 de Março de 2010.

Agradecimentos

No decorrer do mestrado em engenharia de produção da UFSM, tenho muito a quem agradecer.

Primeiramente a Deus, pela vida.

A CAPES pelo fomento a pesquisa, fundamental para o desenvolvimento dos trabalhos.

Ao meu amigo professor Dr. Raul Ceretta Nunes, pela excelente orientação, pelo exemplo de profissionalismo, competência, caráter e trabalho, muito aprendi e muito me inspirei com suas dicas e testemunho.

A todos os colegas da GMICRO: Francisco (Chico), Renato, Bruno, Tiago, Ricardo, Jaziel, Douglas pela amizade e pelo companheirismo.

A minha esposa Juliane e meus filhos Pedro e Camilly, por me motivarem, por entenderem em abdicar de momentos de convivência em função do meu objetivo de fazer mestrado.

Aos meus pais, Juarez e Eliza e meus irmãos, Jailton, Ana e Anelise, pelo apoio e bons conselhos nos momentos difíceis.

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia de Produção
Universidade Federal de Santa Maria

CONTROLE DE ACESSO A RECURSOS COMPUTACIONAIS DE FORMA FLEXÍVEL E DINÂMICA ATRAVÉS DE CONTEXTO

AUTOR: JUNIOR MARCOS BANDEIRA
ORIENTADOR: DR. RAUL CERETTA NUNES
Data e Local da Defesa: 30 de março de 2010, Santa Maria

Um modelo de controle de acesso tem o objetivo de limitar as ações que um usuário legítimo pode exercer em um sistema. Sua finalidade é melhorar a segurança garantindo as propriedades de integridade e confidencialidade. O modelo de controle de acesso considerado padrão é o baseado em perfis ou papéis. Os perfis são funções que o usuário pode exercer e as permissões de acesso a objetos são associadas a um perfil de acordo com a sua função. O modelo baseado em perfis não consegue levar em conta aspectos do ambiente onde ocorre o acesso, isso limita as possibilidades de construção de políticas de segurança mais abrangentes que acompanhem o cenário da evolução tecnológica. Modelos que estendem do modelo baseado em perfis foram propostos, neles existem definições que tornam possível o mapeamento do ambiente onde ocorre o acesso, também chamado de contexto, porém, não existe um consenso em relação à representação do contexto e as políticas de segurança demonstradas pelas definições desses modelos são específicas para seus casos de uso. Esse trabalho apresenta um modelo de controle de acesso baseado em expressões de contexto CABEC. Tal modelo possui definições baseadas nos modelos que estendem o modelo baseado em perfis, porém, demonstra a construção de políticas de segurança para domínios diferentes. As políticas de segurança construídas com o CABEC levam em consideração informações dinâmicas do ambiente e suas combinações. Esses aspectos são importantes, pois aumentam a riqueza das políticas de segurança e a flexibilidade na sua construção. Os aspectos dinâmicos do contexto se referem a informações do instante da interação, são dados como intervalos de tempo, quantidade de acessos simultâneos, meio físico de acesso, localização. O aspecto da flexibilidade na construção das políticas vem da possibilidade do gerente de segurança escolher a quantidade de regras e suas combinações, pode construir uma política que leve em conta a regra de horário de expediente combinada com a função que o sujeito está exercendo naquele momento, ou simplesmente considerar uma regra que leva em conta somente sua localização. Com um modelo de controle de acesso que leva em conta informações dinâmicas e suas combinações aumenta-se a segurança e, conseqüentemente, se ganha em produtividade em relação à propriedade de disponibilidade de serviços, dados e recursos computacionais.

Palavras chave: Controle de Acesso; Contexto; Segurança;

ABSTRACT

Master's Dissertation
Program of Master Degree in Production Engineering
Federal University of Santa Maria

ACCESS CONTROL TO COMPUTER RESOURCES IN A FLEXIBLE AND DYNAMIC THROUGH CONTEXT

AUTHOR: JUNIOR MARCOS BANDEIRA
ADVISER: DR. RAUL CERETTA NUNES
Date and Local: March 30, 2010, Santa Maria

A model of access control aims to limit the actions that a User can have legitimate in a system. Its purpose is to improve security by ensuring the properties of integrity and confidentiality. The model of access control is considered standard based on profiles or roles. The profiles are functions that the User can exercise and access permissions to objects are associated with a profile according to their function. The model-based profiles can not take into account aspects of the environment where the access occurs, it limits the possibilities of establishing policies for the most comprehensive security accompanying the scenario of technological change. Models that extend the model-based profiles have been proposed, there are settings in them that make possible the mapping of the environment where the access occurs, also called context, however, there is no consensus on the representation of context and security policies demonstrated by definitions of these models are specific to their use cases. This work presents a model based access control in context expressions CABEC. This model has definitions based on models that extend the model based on profiles, however, shows the construction of security policies for different domains. Security policies built with the CABEC take into account dynamic information environment and their combinations. These aspects are important because they increase the wealth of security policies and flexibility in its construction. The dynamic aspects of context refers to information the moment of interaction, are given as time intervals, number of simultaneous access, physical access, location. The aspect of flexibility in policy construction comes from the possibility of the security manager to choose the amount of rules and combinations thereof, can build a policy that takes into account the rule of office hours combined with the role that the subject is exerting at the time, or simply considering a rule that takes into account only its location. With a model of access control that takes into account dynamic information and their combination increases the security and, consequently, the gain in productivity in relation to ownership of availability of services, data and computing resources.

***Keywords**: Access control, context, security.*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

O objetivo do controle de acesso é limitar ações que um usuário legítimo de um sistema de computação pode realizar em função das autorizações aplicáveis ao mesmo no momento do acesso (SANDHU & SAMARATI, 1994). Autorizações estabelecem que direitos um sujeito possui em relação a um determinado objeto ou recurso computacional. Controlando o acesso busca-se limitar o uso de objetos de acordo com as autorizações e impedir violações de segurança. As políticas de segurança definem o que pode ou não ser acessado e são impostas diretamente as ações do usuário.

Uma política de controle de acesso define diretrizes de alto nível que determinam como o acesso é controlado e como as decisões de autorização de acesso são estabelecidas (FERRAIOLO et al. 2001). Em outras palavras determinam como um objeto ou recurso computacional será acessado em um sistema computacional. As informações de um sistema computacional só estarão seguras na medida em sejam cumpridas as políticas de segurança que controlam o acesso aos seus dados. Por exemplo, em uma empresa se os dados confidencias dos clientes forem regulados por uma política que determina que somente os gerentes possam ter acesso a estes dados e os demais funcionários não, pode-se dizer que o sistema está minimizando as chances de ocorrência de uma falha de segurança.

Uma entidade (usuário, processo ou máquina) que ganha acesso a recursos de um sistema computacional de forma ilícita, violando uma política de controle de acesso é denominada intruso. As violações de segurança se traduzem como sendo a arte de burlar de alguma forma a política de modo a passar despercebida uma ou mais propriedades de segurança existentes.

Tipo de Violação	***Propriedades de segurança violadas***
Revelação não autorizada	*Confidencialidade*
Modificação não autorizada	*Integridade*
Negação de serviço	*Disponibilidade*

Tabela 1: Relação de violações as propriedades de segurança

A tabela 1 mostra os tipos de violação em contraposição às propriedades de segurança não verificadas (AMOROSO, 1994). As violações de confidencialidade e integridade podem ser evitadas com o controle de acesso e tem impacto na disponibilidade. Confidencialidade significa não revelar dados a pessoas não autorizadas, ou seja, a certas pessoas determinadas informações não dizem respeito e, portanto, devem ser mantidas em sigilo. Integridade significa não permitir a alteração de forma indevida, ou seja, por alguém que não possui as credenciais para tal tarefa. Juntas a confidencialidade e integridade potencializam a disponibilidade, pois minimizam a chance de negação de serviço por acesso indevido seguido de ação maliciosa.

1.1. Motivação

O objetivo desse trabalho é apresentar o Modelo de Controle de Acesso Baseado em Expressões de Contexto (CABEC). O modelo define elementos capazes de considerar aspectos dinâmicos do ambiente que vão além do papel exercido pelo usuário naquele momento. Possibilita a combinação dos elementos na construção de regras para formação de políticas de segurança mais abrangentes e demonstra a criação de regras para mais de um domínio. Além disso, outro objetivo é utilizar padrões abertos na implementação do modelo, para permitir que ele possa ser utilizado pelo maior número possível de sistemas, independentemente da tecnologia na qual foram constituídos.

Modelos de controle de acesso definem uma linguagem para expressar políticas e técnicas que representam uma política de acesso de alto nível. Em geral, não há modelos melhores que os outros, mas sim, modelos mais ou menos adequados ao tipo de política que uma instituição deseja para acesso aos seus recursos computacionais.

Existe uma série de modelos de controle de acesso disponíveis, dentre os quais o RBAC (*Role Based Access Control)* (FERRAIOLO et al., 2001) pode ser considerado padrão de referência. Esse modelo prevê um mecanismo de acesso baseado em perfis (*role)* no qual cada usuário do sistema é associado a um ou a mais perfis (também conhecidos como funções ou papéis) que indicam suas responsabilidades dentro da organização, essa associação é feita quando o usuário é cadastrado e pode ser alterada pelo gerente de segurança quando for o caso. O problema desse modelo é que essas políticas ficam limitadas em considerar apenas o aspecto estático que se refere ao Perfil. Não existe a possibilidade de considerar outras

informações do ambiente como a localização do usuário naquele momento ou a quantidade de acessos simultâneos em uma política de segurança. Esses dados relativos ao contexto podem ser relevantes, pois nem sempre é interessante permitir o acesso de qualquer lugar ou, no caso de acessos simultâneos, limitar a quantidade de acessos, pois podem congestionar uma rede e tornar determinado serviço indisponível.

Já há algum tempo é possível notar a utilização de aspectos dinâmicos na construção de regras de autorização, o *contexto*, mesmo ainda esse termo não sendo bem entendido e explicitado (MCDANIEL,2003). Por exemplo, a figura 01 define um contexto associado com a fila de impressão. Esse contexto revela o estado da fila de impressão. O *Local_manager* verifica se a fila de impressão é menor do que 20% de sua capacidade máxima. O tamanho da fila varia o que torna a aplicação da política mais complexa. Assim, a política de autorização é usada para regular a fila de impressão utilizando uma informação contextual do ambiente, o que traz a vantagem de avaliar um atributo dinâmico, pois o tamanho da fila varia conforme a demanda dos usuários. Como resultado tem-se uma política mais abrangente.

Token_Type	*Printer_load*
Defining Authority	*Local_manager*
Value:	*<20%*

Figura 1 – Controle de acesso a fila de impressão.

O contexto começou a ser considerado um elemento interessante a ser representado nos modelos de controle de acesso. Com o avanço de ambientes de computação pervasiva o mesmo se tornou mais compreendido e é um aspecto central caracterizado pela habilidade de adaptação e realização de tarefas baseadas nas condições ambientais de forma correta e com comportamento adequado (KULKARNI et al., 2008).

As condições ambientais são expressões de *informações do ambiente,* tais como atributos temporais, número de acessos, localidade do recurso ou usuário, entre outras. Elas representam as *características dinâmicas da informação,* uma vez que seu conteúdo pode ser alterado ao longo do tempo.

No sentido de considerar informações ambientais, vários esforços têm sido feitos para aperfeiçoar o modelo RBAC (BERTINO 2001, GEORGIADIS 2001, BACON 2002, JOSHI 2003, MOTTA 2003, COVINGTON 2006, DAMIANI 2006). Os modelos que estendem o

RBAC trazem a agregação do contexto como solução para construção de políticas. O contexto nesses modelos diz respeito a informações dinâmicas do ambiente para domínios determinados. Por exemplo, o TRBAC adiciona atributos temporais associados com perfis que devem ser ativados de acordo com horários pré-especificados, enquanto, o GEO-RBAC leva em conta informações sobre a localização das pessoas baseado em coordenadas geográficas.

1.2. Metodologia

A metodologia utilizada teve cunho teórico reflexivo com bases no mapeamento e análise da literatura sobre: Modelos de controle de acesso, políticas de segurança, contexto, modelos baseados em contexto, tecnologias de baixo acoplamento.

O processo de engenharia de software utilizado para a implementação do modelo e da arquitetura proposta foi o desenvolvimento evolucionário (SOMMERVILLE, 1996), que prevê que as atividades de especificação, projeto, implementação e validação ocorram juntas. Com base em uma especificação inicial, projeta-se e implementa-se um protótipo inicial, para posteriormente ser validado.

A aplicação prática da implementação do Modelo de Controle de Acesso Baseado em Expressões de Contexto foi feita com a sua integração ao banco de dados de geodesastres do Instituto Nacional de Pesquisas Espaciais (INPE), desenvolvido no Centro Regional Sul (CRS/INPE) em Santa Maria, com o objetivo de comprovar a criação de políticas flexíveis e dinâmicas dentro da realidade daquela instituição. A comprovação foi demonstrada através da construção de políticas que levam em conta aspectos variados do contexto.

1.3. Contribuições

O estudo proporcionado por esta pesquisa acerca de controle de acesso baseado em contexto teve a finalidade de estender o modelo tradicional baseado em Perfis com a possibilidade de torná-lo mais abrangente e adequado a realidade atual dos sistemas computacionais, sendo assim, gerou as seguintes contribuições:

- um modelo de controle de acesso que proporciona flexibilidade na construção de políticas, uma vez que é possível a combinação de aspectos relacionados ao contexto para sua construção;

- um modelo que considera informações dinâmicas no controle de acesso; e

- um serviço de controle de acesso de baixo acoplamento, em função de ter sido implementado com a utilização de padrões abertos.

1.2. Organização

Esse trabalho está dividido em seis capítulos. No capitulo 2 é apresentada uma revisão bibliográfica sobre os modelos de controle de acesso, desde os mais tradicionais até os mais atuais que utilizam o contexto como elemento determinante no controle de acesso. No capitulo 3 é apresentada a especificação e a modelagem do Modelo de Controle de Acesso Baseado em Expressões de Contexto (CABEC). No capitulo 4 é apresentada a sua arquitetura e as tecnologias utilizadas para sua implementação. No capitulo 5 é apresentada uma discussão comparativa entre os modelos estudados e o modelo proposto, assim como sua integração com o banco de dados de geodesastres do CRS/INPE, juntamente com detalhes dos testes executados e, por fim, no capitulo 6 são relatadas as conclusões, as contribuições e trabalhos futuros.

2. MODELOS DE CONTROLE DE ACESSO

Diferentes modelos de controle de acesso foram propostos para suportar políticas distintas com objetivo de obter a proteção desejada. Os modelos mais tradicionais de controle de acesso são: Discriminatório (DAC), Obrigatório/Mandatório (MAC) e o modelo baseado em Papéis (RBAC). Destes, o de maior destaque é o RBAC, o qual foi alvo de várias extensões. Neste capítulo os modelos tradicionais são discutidos nas sessões 2.1, 2.2 e 2.3. Logo na sessão 2.4 são apresentados os modelos descendentes do RBAC que consideram contexto nas suas políticas de controle de acesso. Esses modelos são referência para a definição do CABEC.

2.1. Modelo de Controle de Acesso Discriminatório DAC

Neste modelo as autorizações são definidas em listas de controle de acesso localizadas no objeto protegido (SANDHU & SAMARATI, 1994). O conjunto de listas forma uma matriz, onde as linhas correspondem aos sujeitos e as colunas correspondem aos objetos do sistema, conforme pode ser observado na tabela 2. A intersecção das linhas e colunas forma a célula que especifica os atributos de acesso do respectivo sujeito em relação ao objeto considerado.

	O1	**O2**
S1	(read)	(read, write, execute)
S2	-	(write)
S3	(read,execute)	-

Tabela 2 - Modelo de matriz de acesso

A posse pelos usuários dos objetos protegidos caracteriza o controle de acesso discriminatório, o acesso ao objeto é controlado com base na identidade do usuário e nas autorizações que especificam seus privilégios (ler,escrever,executar). As autorizações podem conceder privilégios para um usuário específico ou para um grupo de usuários. Quando um

sujeito tenta efetuar uma operação em um objeto, verifica-se na lista de controle de acesso do objeto se existe alguma autorização que concede o privilégio. Existindo a autorização de acesso o mesmo é concedido, ao contrário é negado.

O proprietário do objeto é livre para conceder ou revogar privilégios de acesso aos outros usuários (JOSHI et al., 2001). Porém, também permite que o usuário autorizado a acessar um objeto faça uma cópia do mesmo e repasse para outros sem permissão explícita do proprietário (SANDHU & SAMARATI, 1994). Permitir que o proprietário do objeto decida quem pode ou não acessá-lo nem sempre é seguro, pois pode conceder direitos de acesso a pessoas não autorizadas pela organização, ao mesmo tempo o modelo não reflete a hierarquia organizacional e dificulta a gerencia de políticas, pois, caso um usuário troque de função é necessária a revogação de todos os seus privilégios antigos e a associação do mesmo com privilégios sobre objetos que a nova função exige.

2.2. Modelo de Controle de Acesso Obrigatório (Mandatório) MAC

O modelo obrigatório baseia-se na classificação de sujeitos e objetos (JOSHI et al., 2001). O nível de segurança associado ao objeto reflete o nível de importância da informação contida no objeto, classificando o objeto quanto ao dano potencial que um acesso não autorizado pode proporcionar (SANDHU & SAMARATI, 1994). A classificação da informação é determinada pelas regras do modelo que definem uma espécie de fluxo dos níveis mais baixos para os mais altos. A tabela 3 mostra as classes historicamente usadas nos âmbitos militares e governamentais.

Ultra secreto
Secreto
Confidencial
Liberado

Tabela 3 - Classes do MAC

Os acessos de sujeitos a objetos somente são liberados quando relacionamentos específicos são satisfeitos entre as classes associadas a ambos, normalmente os seguintes:
1 – A classe do sujeito deve dominar a classe do objeto que ele pretende ler;
2 – A classe do sujeito deve ser dominada pela classe que ele pretende usar para gravar (SANDHU & SAMARATI, 1994).

Considerando um sujeito da classe *secreto*, o mesmo pode ler qualquer objeto da classe igual ou inferior a sua, mas quando grava informação criada, essa só pode ser lida de uma classe igual ou superior a sua. Informações gravadas por um sujeito da classe secreto não podem ser repassadas para usuários de uma classe de acesso menos privilegiada, mesmo quando se fazem cópias da informação. As classes e relacionamentos determinam confidencialidade. Outras classes e relacionamentos podem ser estabelecidos para oferecer integridade e minimizar os conflitos de interesse.

Esse modelo é adequado para algumas organizações, porém, nem todas seguem a hierarquia sugerida. Além disso, o fato de colocar um usuário associado há um dos elementos da hierarquia não corresponde exatamente aos privilégios relativos à função por ele exercida, é possível que várias funções estejam embutidas em um mesmo nível o que pode tornar os objetos acessíveis por membros que não exatamente tem a necessidade de saber de tais informações.

2.3. Modelo de Controle de Acesso Baseado em Papéis RBAC

No modelo de controle de acesso baseado em papéis RBAC permissões/autorizações são dadas a papéis que representam os cargos, funções existentes dentro de uma organização (SANDHU & SAMARATI, 1994). Os papéis representam funções que determinam a autoridade e a responsabilidade concedidas aos usuários (SANDHU e et. al, 1996). A idéia principal é que um usuário pode desempenhar diversos papéis em um sistema, um papel pode ser definido como um conjunto de atividades que estão relacionadas há algum cargo. Esse tipo de controle facilita a gerência de autorização, pois quando um usuário muda de cargo a manutenção das permissões relacionadas aquele cargo ou papel não sofrem mudanças, simplesmente o usuário será desassociado de um papel e associado a outro.

O trabalho apresentado ao NIST *National Institute of Standards and Tecnology*, denominado NIST RBAC (FERRAIOLO et al., 2001) define uma padronização para o modelo que é organizada em quatro passos seqüenciais incrementados funcionalmente, esses níveis são cumulativos o que significa que cada um inclui os requerimentos previstos nos anteriores, são eles: Básico, Hierárquico, Com Restrições e Simétrico.

2.3.1. RBAC básico

Define a estrutura essencial para que um modelo seja considerado baseado em papéis, composto pelos elementos: usuário, papéis e permissões com suas respectivas semânticas, como pode ser visto na figura 2. O essencial é que usuários sejam associados a papéis e que

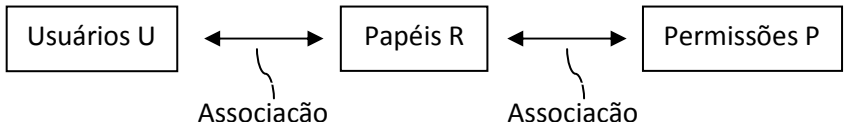

Figura 2 - Modelo RBAC básico

os mesmos já estejam associados à permissões/autorizações. Ainda possibilita que as associações usuário-papel e papel-permissão sejam de muitos para muitos e que um usuário após iniciada uma sessão possa ativar/desativar mais de um papel ao mesmo tempo.

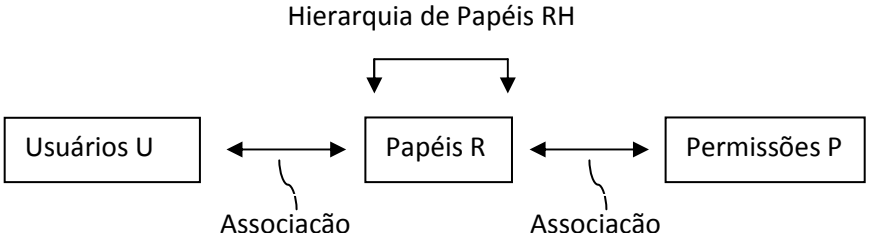

Figura 3 - RBAC Hierárquico

2.3.2. RBAC Hierárquico

Incorpora ao RBAC básico requisitos para suportar a hierarquia de papéis conforme pode ser notado na figura 3. Matematicamente uma hierarquia é uma ordem parcial definindo uma relação de precedência de papéis, por meio da qual, papéis de categoria superior adquirem as permissões de seus subordinados. Dessa forma é possível representar a hierarquia funcional de uma organização onde papéis de maior responsabilidade, mais específicos, acabam incorporando autorizações de papéis de menor responsabilidade.

2.3.3. RBAC com Restrições

Restrições RBAC adicionam requerimentos que forçam a separação de responsabilidades relacionadas a conflitos de interesse. Conflitos de interesse se referem ao fato de um usuário ganhar autorização para permissões associadas com conflito de funções. Por exemplo, um usuário não pode ter o papel Supervisor de Caixa e Caixa ao mesmo tempo, pois isso leva ao risco de ocorrência de fraudes ou erros na manipulação da informação. Existem duas formas de se obter a separação de responsabilidades: Estática e Dinâmica.

Na Separação de Responsabilidade Estática (SRE) as restrições são impostas no momento da atribuição de papéis para usuários na medida em que são previstos conflitos de interesse em relação ao exercício de funções mutuamente exclusivas. Um usuário que é Supervisor de Caixa não pode ser Caixa, pois as duas funções são exclusivas.

Com Separação de Responsabilidade Dinâmica (SRD) o usuário pode ser atribuído a duas funções exclusivas, desde que exerça as funções separadamente, no momento que tentar acionar as duas simultaneamente é que a política de conflito de interesse deve ser verificada. Um usuário pode ser autorizado para ambas às funções, Caixa e Supervisor de Caixa, desde que quando esteja exercendo uma não consiga exercer a outra função. Para isso deve abandonar a função anterior e se autenticar com a nova função. A adição de SRE e SRD na estrutura do modelo RBAC pode ser observada na figura 4.

2.3.4. RBAC Simétrico

Neste último nível do modelo RBAC adiciona-se o requisito ao suporte a revisão de associações permissão-papel similar a revisão usuário-papel do RBAC básico. Isso se torna necessário em função de que as associações entre papel-permissão podem se tornar inadequadas com o passar do tempo, principalmente onde são atribuídos em ambientes de administração distribuída com o esforço coordenado de múltiplos administradores. A interface do RBAC simétrico deve possibilitar o retorno de todos associados a permissões de um perfil, ou o conjunto completo de operações e pares de objetos associados a permissões associadas a um usuário em particular ou perfil e também permitir a definição direta e indireta de permissões. As diretas são atribuídas diretamente para um usuário ou perfil enquanto que as indiretas são as herdadas.

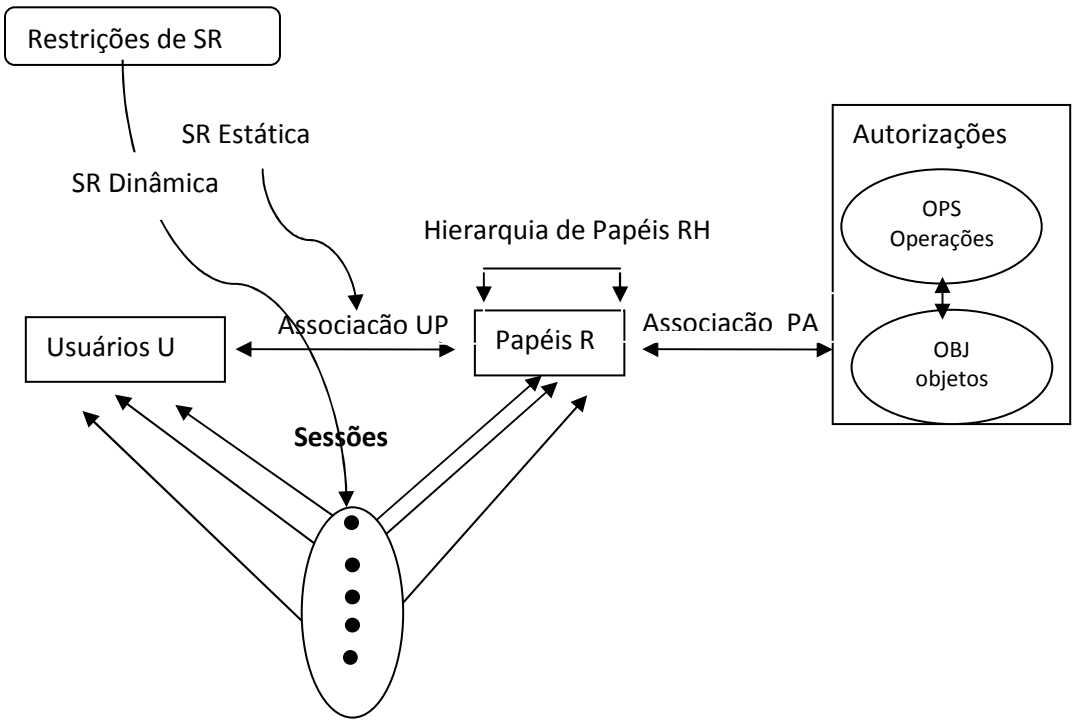

Figura 4 - RBAC com Restrições

2.3.5. Vantagens do RBAC

O controle de acesso baseado em papéis tem recebido bastante atenção como um enfoque generalizado para o controle de acesso, pois incorpora reconhecidas vantagens (JOSHI et al., 2001):

- É politicamente neutro (OSBORN et. al, 2001), ou seja, expressa diferentes políticas de acesso em vez de embutir uma política particular como no DAC.
- Pode ser configurado para suportar tanto políticas internas, quanto as tradicionais dos modelos MAC e DAC.
- Mapeia o organograma organizacional de modo a permitir mais flexibilidade, impessoalidade e simplicidade na construção de políticas de segurança para controle de acesso.
- Permite o conceito de privilégio mínimo, fazendo com que o usuário execute operações que suas funções sugerem, evitando assim ações potencialmente danosas.
- Suporta o conceito de separação de responsabilidades, evitando-se a concentração de poder em um único usuário e reduzindo as chances de fraudes ou danos acidentais.

As vantagens mencionadas vêm ao encontro das motivações dos propositores do RBAC: simplificar a administração de políticas de segurança e ao mesmo tempo facilitar a definição de políticas flexíveis e customizadas (FERRIOLO et al., 2001). O RBAC se firmou como modelo genérico, porém, possui deficiências em relação aos novos cenários proporcionados pelo do avanço tecnológico atual, ele não consegue levar em consideração aspectos dinâmicos no controle de acesso, apenas informações estáticas que são relatadas antes dos usuários interagirem com o sistema. Não há, por exemplo, como restringir o acesso a um usuário que tenha o perfil Administrador de Rede, mas esteja acessando o servidor de uma rede externa durante a madrugada, o que, dependendo das políticas de segurança de determinada instituição pode não ser interessante. Tendo em vista as deficiências do RBAC pesquisas têm sido feitas levando em conta a possibilidade de adaptações com extensões do mesmo de acordo com a realidade que certas instituições que exigem políticas mais abrangentes. As pesquisas mais relevantes sobre novos modelos descendentes do RBAC são mencionadas a partir da próxima sessão.

2.4. Modelos descendentes do RBAC

Muitos modelos de controle de acesso que estendem o RBAC foram propostos pela literatura (BACON *et al.,* 2002; BERTINO *et al.*, 2001; COVINGTON *et al.,* 2001; COVINGTON *et al.* 2003; GEORGIADIS *et al.,* 2001; JOSHI *et al.*, 2003; KUMAR *et al.,2003;* LIN & BROWN, 2000; LONGSTAF, 2003, COVINGTON, 2006, SOARES, 2006, KULKARNI D. et al., 2008). Todos esses modelos, com exceção de (DAMIANI *et al.*, 2006), (COVINGTON, 2006), (SOARES, 2006), (KULKARNI D. et al., 2008), são revisados por (MOTTA, 2003) o qual faz uma sumarização de alguns deles onde caracteriza e diferencia-os quanto a profundidade de cada um em relação aos quatro níveis do RBAC, uso de regras contextuais, implementação e aplicabilidade no âmbito de organizações de saúde que utilizam alguns destes para controle de acesso ao Prontuário Eletrônico de Pacientes PEP.

Examinando tais modelos, foram escolhidos os mais representativos em relação ao uso de contexto na sua modelagem para uma explicação mais detalhada, a ordem das descrições é cronológica para que seja possível a noção da evolução dos mesmos com o passar do tempo.

2.4.1. TRBAC

Um dos primeiros trabalhos a estender o tradicional modelo de controle de acesso baseado em perfis RBAC e introduzir informações de contexto com a utilização de atributos temporais como condição de habilitação do Perfil foi o TRBAC (*Temporal Role-Based Access Control*) (BERTINO, et al., 2001). Nesse modelo os perfis são habilitados dentro de seus turnos, por exemplo, o perfil *medico-turno-noite* será habilitado no turno da noite. O modelo utiliza o que chama de *Role Enabled Based* (REB) que é constituído de um intervalo de tempo, uma expressão periódica e uma expressão de prioridade. Cada linha do REB é formada por um Evento Periódico com esses elementos,

(PE_1) . ([1/1/2000,∞], turno-noite, VH:habilite medico-turno-noite

Esse é um exemplo para o domínio médico, onde o evento periódico PE_1 indica um intervalo de tempo que tem data de inicio em *1/1/2000 sem data final* para uma expressão periódica chamada *turno-noite* com prioridade *Muito alta (VH)* para habilitação do perfil *medico-turno-noite*. Um aperfeiçoamento desse modelo resultou no GTRBAC (*Generalized*

Temporal Role-Based Access Control) (JOSHI et al., 2003), nele existe uma separação entre a condição de perfil habilitado e perfil ativo, ou seja, um perfil pode estar habilitado dentro do seu turno, porém ainda deve obedecer a uma restrição para ser ativado, essa restrição pode ser relacionada aos dias nos quais está escalado para trabalhar, por exemplo, a restrição ((*S,Q,X)*, *assign*Joao*to*MedicoPlantonista) indica que o usuário João somente será associado ao papel *MedicoPlantonista* na segunda-feira (S), na quarta-feira (Q) e na sexta-feira (X), dias dos seus plantões. Como se pode perceber nesses modelos já é possível notar mais abrangência na construção de regras de autorização, pois não basta, por exemplo, o sujeito ter o perfil de médico para possuir todas as permissões cabíveis a sua função, é necessário também que esteja no seu turno e dentro dos dias nos quais está habilitado para exercer esses turnos. Com as informações contextuais torna-se possível a construção de tal regra.

As condições de habilitação e ativação dos perfis são armazenadas em tabelas e avaliadas com o uso de gatilhos disparados por um Sistema de Gerenciamento de Banco de Dados no momento em que alguma ação é solicitada, caso o usuário tenha as condições temporais favoráveis ao atendimento da regra, o acesso é concedido, do contrário não.

2.4.2. E-RBAC

O modelo E-RBAC (*Environment Role-Based Access Control)* foi desenvolvido por (COVINGTON et al., 2001) com o objetivo de ativar Papéis ambientais no controle de acesso a residências. O papel ambiental não é vinculado ao usuário, mas sim a um conjunto de autorizações e uma regra estabelece as condições de sua ativação/desativação. As informações são capturadas por sensores espalhados pela residência que indicam horário, temperatura, nível de ruído, localização do usuário, etc.

O acesso para executar uma operação de um objeto é concedido a um usuário quando ele é membro do papel tradicional e todos os papéis, do conjunto de papéis ambientais estão ativos no momento da solicitação da autorização. Por exemplo: *{Criança, Assistir, Televisão,{Sala de Estar Ocupada, Horário de Lazer},+}* estabelece que indivíduos com o papel tradicional criança somente podem assistir televisão quando os papéis ambientais *Sala de Estar Ocupada* e *Horário de Lazer* estiverem ativos, ou seja, quando efetivamente a sala de estar estiver ocupada e o horário for o de lazer.

No E-RBAC existe suporte para hierarquia geral de perfis tradicionais e convencionais, também separação dinâmica de responsabilidades. É relatado ainda que as autorizações possam ser positivas ou negativas, porém, os autores não mencionam como resolveriam os conflitos quando, para um mesmo individuo, uma autorização concede o acesso e outra proíbe. O contexto é suportado pelo modelo na definição das condições de ativação e desativação dos perfis ambientais. O modelo foi implementado em forma de protótipo na plataforma Java 2 SE.

2.4.3. XORBAC

O modelo XORBAC (*eXtended Object Role-Based Access Control*) foi desenvolvido por (NEUMANN e STREMBECK, 2003). Este modelo estende o RBAC básico com a associação de restrições contextuais aos papéis de usuário para garantir uma autorização de acesso. Uma restrição contextual é formada por um conjunto de condições contextuais que possuem atributos relacionados ao ambiente de onde se requer a autorização. Os valores dos atributos são obtidos por chamadas a sensores lógicos ou físicos que representam propriedades ambientais que mudam dinamicamente.

As propriedades contextuais podem variar dinamicamente (hora, data) ou de acordo com a instância da entidade representada (nome do usuário, data de aniversário do usuário, local de acesso). Para exemplificar como as restrições contextuais são associadas a papéis na construção de regras de acesso, considere que um sistema de testes de um portal educacional da Internet poderia estabelecer que um estudante somente pudesse realizar os testes designados para ele, na data marcada, no horário comercial, e a partir de estações com certo domínio de endereço IP preestabelecidos. A restrição contextual seria: *{cond1: matricula_estudante(usuário) == matricula_exame(teste), cond2: data_atual() == data_exame(teste), cond3: hora_atual() in horário_comercial(), cond4: IP_cliente() == *.universidade.edu.br}* à autorização <Realizar, Teste> submete a realização de testes se todas as condições forem verdadeiras, assim temos uma autorização condicional, essa autorização será associada ao papel *Estudante.*

No XORBAC existe suporte para hierarquia geral de papéis com herança de autorizações e a separação estática de responsabilidades pela definição de papéis mutuamente exclusivos ou de autorizações mutuamente exclusivas. Dois ou mais papéis em exclusão

mútua não admitem usuários em comum, enquanto que duas ou mais autorizações mutuamente exclusivas não podem ser associadas ao mesmo papel. O modelo foi implementado em XOTcl, uma extensão orientada a objetos da linguagem de programação Tcl e está disponível no endereço: *HTTP://wi.wu-wien.ac.at/home/mark/xoRBAC/index.html* com uma versão para o sistema operacional Windows e outra para Linux. Os autores relatam a necessidade de realização de mais testes com o modelo em outros domínios, embora defendam que já foi adquirida boa experiência em relação ao potencial do mesmo em ambientes colaborativos na Web.

2.4.4. MACA

Outro modelo que estende o RBAC com a utilização de contexto é o MACA (*Middleware* de Autenticação e Controle de Acesso) (MOTTA, 2003), o qual leva em conta contexto nas regras de autorização representando-o por uma expressão lógica denominada *regra de autorização*, avaliado no momento em que o usuário tenta acessar um objeto em uma sessão. A expressão lógica constitui-se de informações do ambiente de onde ocorre o acesso, como: localização, data, hora. Essas informações são consideradas *variáveis de contexto* e são associadas a *Entidades*. Uma entidade é algo que pode ser caracterizado e relevante para interação entre usuário e aplicação.

Cada expressão lógica é uma associação de contextos, no MACA são definidos os contextos: Usuário (*UsuarioCtx)*, Data (*DtCtx)*, Rede (*netCtx)* e Paciente (*pacCtx).* O contexto classifica as informações de acordo com o seu significado, *DtCtx* agrega dados relativos a data como dia, ano, mês. Uma expressão lógica combina contextos com operadores lógicos para formação de regras. Fazendo uma analogia com o CABEC, pode-se dizer que a expressão lógica no MACA é algo semelhante a uma expressão de contexto no CABEC. Para representação das expressões lógicas foram definidos três componentes:

Variável Contextual: Valor obtido do ambiente na tentativa de acesso associado a m contexto. Exemplo de variáveis: *usuarioCtx.nome*, *dtCtx.data*, *netCtx.peer_ip*.

Conjunto Contextual: Conjunto de valores determinados e diferenciáveis encontrados no ambiente na tentativa de acesso. Exemplos: Dias úteis da semana, Estações de trabalho da sala

de emergência, pacientes internados. Exemplo de variáveis: *dtCtx.dias_uteis, pacCtx.pacientes_internados, netCtx.dominios_emergencia.*

Função Contextual: Relação entre as entidades existentes no ambiente de uma tentativa de acesso que não podem ser representadas por variáveis ou conjuntos contextuais. Exemplo: *pacCtx.assistido_por(umCodPac, usuarioCtx.registro_profissional*).

Quando os componentes de contexto que representam informações contextuais são relacionados dentro de expressões lógicas é possível representar políticas de acesso para um objeto protegido. As políticas com suas regras são definidas em uma linguagem de expressões lógicas capaz de recuperar valores provenientes de variáveis, conjuntos e funções de contexto para relacioná-los por meio de operadores aritméticos (+,-,*,/,..), relacionais (>,<,=,!=, in, ...) e booleanos (&,|,!). Exemplo de regra:
umCodPac ***in*** *pacCtx.pacientes_agendados(dtCtx.data)* | *umCodPac* ***in*** *pacCtx.Pacientes_internados*

Tal regra estabelece que um acesso somente seja concedido se o paciente identificado por *umCodPac* está internado ou tem uma consulta agendada na data corrente. Note que a regra consegue mapear dados que vão além da simples associação entre perfis e permissões, não basta ter um perfil Médico com a permissão de Leitura para obter acesso ao prontuário eletrônico do paciente, é preciso que ele esteja com consulta agendada com o mesmo ou que esteja internado no hospital. Assim, o modelo consegue mostrar que através da representação de contexto é possível construir regras de autorização mais abrangentes dentro desse domínio hospitalar.

O MACA foi desenvolvido através de uma implementação integrada ao serviço de diretórios LDAP (*Lightweight Directoty Access Protocol*) com linguagem de programação Java e padrões de segurança da arquitetura CORBA para prover acesso ao Prontuário Eletrônico de Pacientes do Instituto do Coração (InCor) de São Paulo.

2.4.5. GSAM

O Modelo de Autorização para Dados Geo-Espaciais GSAM (VIJAYALAKSHMI e JOON, 2004) diferentemente dos modelos tradicionais que se baseiam em objetos e sujeitos para expressar autorizações, leva em conta *expressões credenciais* e *expressões de objetos*

para representar atributos temporais e espaciais. O modelo apresenta *S={s1,s2,s3...}* como grupo de sujeitos, *O ={o1,o2,o3...}* como grupo de objetos e um conjunto de privilégios *M={ver, aumentar zoom, sobrepor, identificar, baixar-dados, atualizar, inserir, deletar, compor, animar, sobrevoar, ver-thumbs, ver-anotacoes}*. São exemplos de políticas suportadas pelo GSAM:

P1 = Imagens de baixa resolução podem ser vistas por todos.

P2= Um terreno localizado no endereço "rua Presidente Vargas, 123, Santa Maria, RS" pode ser acessado somente pelo seu proprietário atual.

P3= Uma imagem de resolução em metros sobre o Afeganistão somente pode ser disponibilizada para autoridades militares.

P4= Somente os oficiais de polícia do estado do Rio Grande do Sul tem permissão para acessar imagens de resolução de um metro da planta de energia nuclear localizada nas coordenadas 65,45,20,20.

Nota-se que P1 e P2 são políticas associadas à privacidade dos dados e P3, P4 com a segurança nacional.

Para montar essas políticas foram definidas credencias para os **sujeitos** e cada credencial pode ter um conjunto de atributos, por exemplo, a credencial *proprietário* tem os atributos: *endereço, bloco, lote* que podem ser obrigatórios ou opcionais. Um sujeito pode ser associado a mais de uma credencial, pode ser *proprietário* e também *policial*, ou seja, pode possuir um conjunto de credenciais. Para expressar autorizações foram definidas expressões credenciais *CE*, cada expressão pode ser formada por um conjunto de variáveis *CV*= {*A(ct1) U A(ct2) U....* } onde *A(cti)* é o conjunto de atributos associados a cada contexto *cti*. Exemplo de expressão credencial:

Ce1=(proprietário(x)^(endereço_casa=” rua Presidente Vargas, 123, Santa Maria, RS”)^(bloco=”23”)^(lote=”2”)).

Essa expressão denota todos os proprietários no endereço 123 rua Presidente Vargas, Santa Maria RS no bloco 23 e lote 2. No GSAM, políticas também podem ser associadas a **objetos** *geo-espaciais* que incluem imagens de satélite, fotos aéreas, dados digitalizados abrangendo a superfície da terra, dados cartográficos constituídos de pontos, linhas e áreas que constituem as formas e as localizações, tais como edifícios, ruas, ou

cidades. Cada objeto geo-espacial *o* é representado como uma tupla *o=(oid, y,lt,lg, h, w, r, t, l)*, onde *oid* é um identificador único, *y* o tipo de objeto geo-espacial, *lt,lg,h* e *w* a latitude, longitude, altura e largura respectivamente, que representam os limites mínimos da caixa do objeto, *r* a resolução, *t* o tempo (tempo de download, última atualização), e *l* o link temático para o dado associado com o objeto. Com as representações dos objetos geo-espaciais é possível a criação de expressões espaço-temporais:

Ge1: (tipo(x)=imagem^retângulo(x) contem(10,20,10,10) ^ time(x) dentro[fevereiro2:2002,marco1:2002]^resolução(x) <= 10metros)

Essa expressão espaço temporal especifica que uma imagem espacial tem área (10,20,10,10) com download realizado entre 2 de fevereiro de 2002 e primeiro de março de 2002, cuja resolução é maior que 10 metros.

Com as expressões associadas a sujeitos junto com expressões associadas a objetos é possível a criação de autorizações:

a1= (Jhon(x), type(y) = LANDSAT ^ retangulo(x)=(50,60,10,10),{zoon-in:8},[1,1,1999,agora]).

Tal autorização especifica: João tem permissão de acesso a região centrada no ponto (50,60) com largura e altura 10 dentro em imagens do LANDSAT, com *zoom-in* em nível superior a 8, durante 1 de janeiro de 1999 até a data atual.

O modelo foi implementado e encontra-se disponível no endereço http://cimic.rutgers.edu/spatial. Por ser um modelo que não segue o padrão RBAC não possui referencias relacionadas à hierarquia de perfis e separação de responsabilidades.

2.4.6. GEO-RBAC

O modelo GEO-RBAC proposto em (DAMIANI & BERTINO, 2006) é uma extensão do RBAC com o objetivo de atribuir permissões de acordo com a localização espacial do usuário. É introduzido o conceito de papel espacial e suporte a representações homogenias de todos os aspectos envolvendo papéis, objetos e informações contextuais, assim como a posição de um usuário. O modelo espacial adotado é compatível com o *Open GeoSpatial Consostium* (OGC). Este é baseado na noção de tipo de característica (rua, cidade, região) e

característica, que é uma instância de um dado tipo (rua A10, Milan, Lombardy). Características podem ser definidas geometricamente (representação por pontos, linhas ou polígonos) dentro de uma referência espacial. Objetos no GEO-RBAC correspondem à configuração de características de um dado tipo.

Um papel espacial representa os limites geográficos de uma função organizacional. Os limites são definidos como características: rua, cidade ou um hospital. O limite especifica a extensão espacial na qual o usuário pode ser locado para ser habilitado a exercer um papel. Baseado na posição física obtida através de um terminal móvel com GPS ou um telefone celular, usuários são associados as suas posições lógicas representando a característica na qual o mesmo está locado. Se o usuário está localizado dentro dos limites espaciais nos quais ele pode exercer determinado papel o qual foi ativado por login durante uma sessão o papel é dito como habilitado. Para especificar o tipo de limite espacial de um papel e a granularidade de sua posição lógica, foi introduzido o conceito de Papel Espacial de Esquema. Papéis espaciais são assim especificados como instâncias de Papéis de Esquema.

A localização espacial é feita de acordo com o OGC, sendo assim, características são ditas espaciais quando podem ser mapeadas em localizações em um dado espaço, como cidade de Milão ou lago Michegan. A localização de uma característica é representada pela geometria. Ao contrário, uma característica não-espacial não está associada a nenhuma localização, por exemplo, um carro identificado pela placa AZ213JW.

Característica tem uma aplicação dependente de sua semântica, que é expressa através do conceito de tipo de característica. O tipo da característica capturada revela o significado da entidade. Rua, cidade, carro e lago são exemplos de tipo de característica.

A idéia central do GEO-RBAC é a distinção entre o conceito de Papel de Esquema e Papel de Instância. Um Papel de Esquema define algumas propriedades comuns de um conjunto de funções organizacionais com um significado similar. Um papel de Esquema não somente define um nome comum para um conjunto de Papéis espaciais, mas também limita o espaço onde Papéis podem ser habilitados. Além disso, especifica o tipo de localização lógica e em última instância a granularidade da posição que o usuário pode exercer o papel o qual está associado. Um Papel de Instância é um papel que satisfaz as restrições definidas no nível de esquema.

Um exemplo de tupla que representa um papel de esquema*: <Doutor, Hospital, Setor, Msetor>*, onde, *Doutor* é o nome do Papel; *Hospital* é o tipo de característica do papel medido, o tipo de objeto que espacialmente contém o Papel; *Setor* é o tipo de característica da

posição lógica, pois é suposto que toda área do hospital é subdividida em setores; finalmente *Msetor* a posição mapeando a função, o setor que contém a real posição do usuário.

O modelo GEO-RBAC também suporta hierarquia de papéis com herança de permissões e restrições de separação de responsabilidades para lidar com características específicas, diferentes granularidades (nível de esquema, nível de instância), dimensão (espacial, não-espacial) e verificação de tempo (estático/ativação dinâmica/habilitação dinâmica).

2.4.7. CIBAC

O CIBAC (*Contextual Information-Based Access Control*) é proposto por (SOARES, 2006) e estende o RBAC básico com a introdução de informações contextuais na regra de autorização de acesso. O modelo utiliza uma *Condição de Contexto,* que é associada a um *Tipo de Contexto (Objeto ou usuário)* e baseada em *Propriedades de Contexto* captadas do ambiente de onde se faz a interação com o sistema, com o objetivo de autorizar o acesso mediante conformidade com as políticas de segurança que mapeiam determinado contexto. Por exemplo, na tupla <CT, P, +, V> CT é um Tipo de Contexto <*Context Type = "Sujeito"*>, P é o nome da propriedade <*Property Name = "Tempo"*>, + é o operador <*Operador OP ="+"*> e V é o valor da propriedade <"*Value V = "22:00"*>. Essa Condição de Contexto especifica que o sujeito só poderá ter acesso depois das 22:00 horas.

O modelo também possibilita a agregação de mais de uma *Condição de Contexto* para especificar mais de uma regra de acesso, assim, pode-se mapear mais detalhadamente o ambiente onde está o usuário.

Segundo (SOARES, 2006) o CIBAC implementa o conceito de hierarquia de papéis, embora o autor não especifique como isso acontece no âmbito do modelo, também existe suporte para delegação de atribuições, através do mapeamento da *Ação* de *delegar* como uma *Condição de Contexto,* de modo que quando a ação é concedida o controle de acesso passa ser *Discriminatório.* Por exemplo, suponha que um usuário "x" deseja delegar uma dada ação sobre o arquivo *Ordem_Médica.doc* ao usuário "y". A responsabilidade do CIBAC é determinar se esta delegação deve ser permitida. Isto é feito através da consulta às condições de contexto para a ação de delegar.

O CIBAC foi desenvolvido em uma *Arquitetura Orientada a Serviços SOA*, conforme pode ser visto na figura 5, de modo que possui três *Web Services* na sua constituição: *Serviço de Administração, Serviço de Decisão* e *Serviço de Autorização:*

*Web Service Administração: C*ontempla o acesso ao banco de dados e provê diversas operações para a manipulação dos mesmos como: Inserir, consultar e apagar.

Web Service Decisão: Responsável pela manipulação dos arquivos contendo as políticas de acesso e recebimento das informações contextuais das requisições de acesso confrontando-as com as diversas políticas previamente cadastradas.

Web Service Autorização: O Serviço de Autorização desempenha a função de gerente do CIBAC, ou seja, coordena todo o processo de autorização para autorizar ou não um determinado sujeito a executar uma ação que envolve algum objeto (recurso). Para que este serviço retorne uma resposta apropriada, ele utiliza-se dos dois serviços descritos anteriormente.

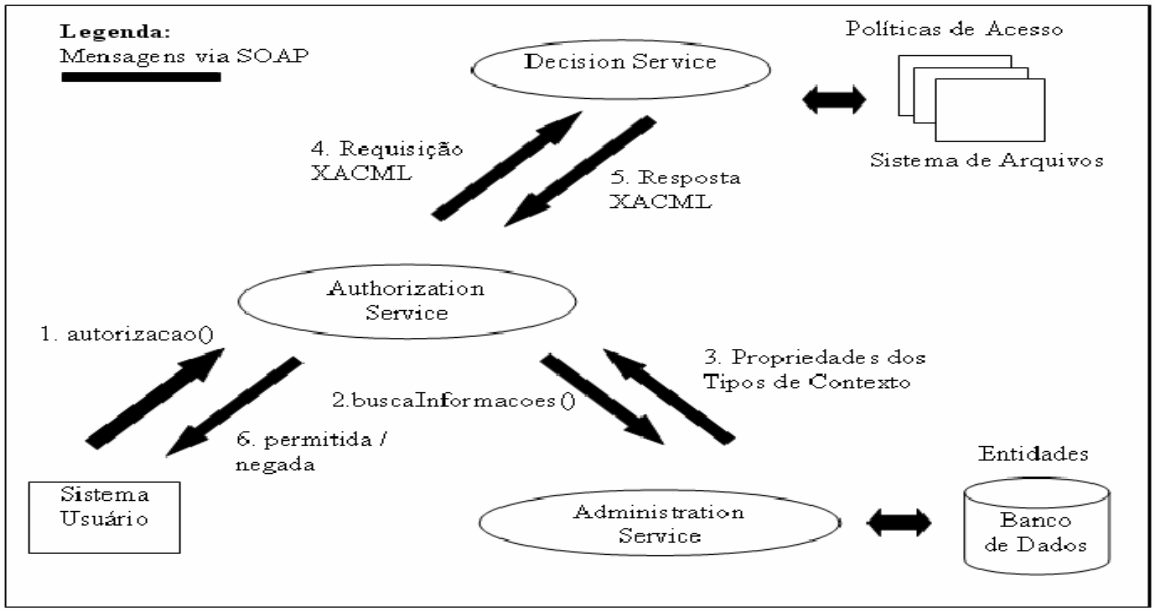

Figura 5 – Arquitetura CIBAC, fonte SOARES 2006

O modelo CIBAC foi proposto e desenvolvido para controlar o acesso aos dados do Prontuário Eletrônico de Paciente PEP do Hospital Universitário de Santa Maria, sendo assim, foi integrado ao Sistema de Gerenciamento e Controle de Aplicações denominado SGCA. O SGCA é uma ferramenta utilizada para controlar o acesso dos usuários às aplicações do sistema SIE (Sistema de Informações Educacionais) implementado pelo CPD/UFSM e que

contém os programas usados pelo hospital e comporta o PEP. O SGCA armazena senhas e garante que cada usuário tenha acesso somente às aplicações pertinentes a ele.

Assim, no SIE para tomar uma decisão, ao invés de consular sua base de dados, o SCGA comunica-se com o CIBAC, através do Serviço de Autorização. Deste modo o modelo discricionário do SCGA pode ser substituído pelo modelo de controle de acesso baseado em informações contextuais do CIBAC.

2.4.8. CA-RBAC

O modelo Context-aware RBAC (KULKARNI D. e TRIPATHI, 2008) é embutido em um framework para programação e projeto de aplicações sensíveis a contexto. A característica nova desse framework é o alto nível de abstração para especificar requerimentos de controle de acesso baseado em contexto especialmente relacionado a alguns aspectos como:

1- *Admissão de Perfil e condições de validação*: São condições de contexto que necessitam ser satisfeitas antes de admitir um usuário a um Perfil e também para ele continuar sendo membro daquele perfil.
2- *Permissões de perfil personalizadas*: Algumas permissões permitem que diferentes membros de um perfil possam acessar diferentes espaços ativos baseados em seus contextos individuais.
3- *Permissões baseadas em contexto ativando condições*: Essas condições são associadas com uma permissão de perfil especifica e condição de contexto especifica que necessitam ser asseguradas para um membro do perfil executar alguma permissão.
4- *Condições de acesso a recursos baseadas em contexto*: Essas condições restringem o acesso de um membro para um subconjunto de recursos que são gerenciados por um serviço de espaço ativo.

O modelo CA-RBAC propõem dois elementos: Camada de gerenciamento de contexto e camada de controle de acesso.

Camada de Gerenciamento de Contexto: Nessa camada atuam os agentes de contexto que se autenticam junto a sensores dos quais irão coletar informações do ambiente. A aplicação necessita confiar no agente do qual irá obter as informações de contexto requeridas, pois elas serão usadas nas decisões de controle de acesso. No modelo CA-RBAC condições de

contexto são expressas como predicados que são avaliados por gerentes de perfil/objeto. Assim, predicados são compostos de consultas suportadas por agentes de contexto e gerenciadores de perfis. Esses predicados são usados pelos gerenciadores de perfil/objeto como parte da política de controle de acesso baseada em contexto. A ocorrência de certas condições de contexto é comunicada pelos agentes de contexto para a camada de controle de acesso através da notificação de eventos de contexto.

Camada de Controle de Acesso: A principal diferença entre o modelo NIST RBAC e o CA-RBAC está no conceito de como as permissões são consideradas. Enquanto no modelo NIST RBAC a aprovação de uma permissão implica na realização de uma ação especifica em um objeto especifico com o mesmo efeito independente do membro do perfil que está executando a ação no CA-RBAC as permissões podem ser personalizadas para cada membro do perfil. As operações de Perfil quando executadas por diferentes membros podem acessar diferentes objetos baseado no contexto individual do membro do perfil, além disso, permissões são associadas a objetos que podem estar ligados a diferentes serviços em um espaço ativo sobre diferentes condições de contexto.

Para suportar permissões personalizadas objetos são classificados como *privados* e *compartilhados.* Compartilhados são objetos comuns para todos os membros de todos os perfis, já objetos privados são específicos para um perfil e são gerenciados separadamente para diferentes membros daquele perfil. As ligações de ambos, objetos privados e compartilhados podem mudar de acordo com as mudanças nas condições de contexto. As permissões personalizadas são especificadas para objetos privados associados há um perfil.

O modelo CA-RBAC tem sido implementado através de um framework em XML, algumas simulações de seu uso em aplicações de computação pervasiva são mencionadas, porém, não existe relato de um teste prático do modelo.

2.5. Conclusões do capítulo

Nesse capítulo foram revisados os principais modelos de controle de acesso com ênfase nos baseados em contexto. Percebe-se que a modelagem utilizada é baseada em definições que correspondem ao domínio do problema em questão. Se o problema é acesso a informações de um determinado domínio (médico, espacial, residencial, etc), são criados elementos para representar as informações contextuais para o domínio específico. Como

resultado, cada modelo é adequado apenas para seu caso de aplicação especifico e dificilmente demonstra como pode ser utilizado em outros casos, podendo não ser utilizado.

A construção de regras de acesso é feita com as definições cada modelo. Agregado as definições estão operadores aritméticos e funções que permitem testar um conjunto de informações relativas a cada contexto para confirmar ou não uma condição de acesso pré-determinada.

3. MODELO DE CONTROLE DE ACESSO BASEADO EM EXPRESSÕES DE CONTEXTO - CABEC

Um modelo de controle de acesso baseado em contexto deve representar as variáveis ambientais que compõem os ambientes de modo geral. Nesse capítulo é apresentado o Modelo de Controle de Acesso Baseado em Contexto (CABEC). Na primeira seção (3.1) é explicado como regras de acesso podem ser representadas através de expressões contextuais. Em seguida, na seção 3.2, define-se a terminologia e definições que compõem o CABEC. A seção 3.3 demonstra como o CABEC estende o RBAC e a seção 3.4 explica como é possível construir e combinar regras que levam em conta aspectos dinâmicos para diferentes domínios. Finalmente a seção 3.5 apresenta um algoritmo para realizar a avaliação dos contextos e a seção 3.6 apresenta as considerações finais deste capítulo.

3.1. Construção de Regras Através de Expressões de Contexto

O CABEC considera o fato de que regras de acesso a recursos e dados computacionais definem políticas de segurança baseadas em contexto, e que essas regras podem ser construídas com *expressões de contexto*. Esse conceito foi referenciado em (VIJAYALAKSHMI e JOON, 2004) que o aborda em forma de Expressões Credenciais e Expressões de objeto, porém, no CABEC as *Expressões de Contexto* não possuem classificação, uma vez que representam informações associadas a *Entidades de Contexto* que são elementos que sofrem ou executam uma ação. As *Expressões de Contexto* retiradas do ambiente devem ser comparadas com *Expressões de Contexto* contidas nas regras com o objetivo de verificar se o contexto estático representado pela regra é o mesmo que as expressões contextuais informam naquele momento e assim conceder ou não o acesso. Um exemplo de como podem ser formadas as *Expressões de Contexto* pode ser observado na figura 6.

As *Expressões de Contexto* podem se tornar uma regra de autorização: *O chefe da UTI tem permissão de acesso no objeto Prontuário Eletrônico desde que esteja acessando o mesmo da própria UTI.* Assim, para conseguir mapear as expressões contextuais a seguir é introduzida a definição formal do CABEC.

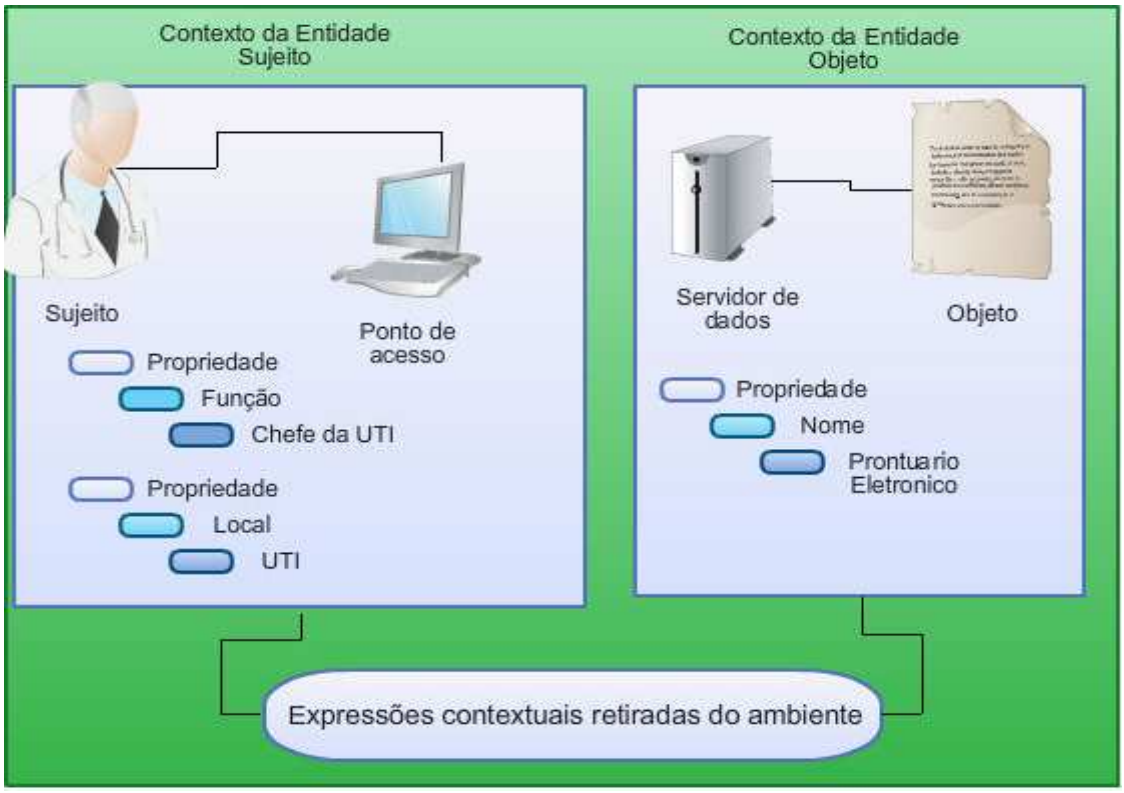

Figura 6 - Exemplo da extração de expressões contextuais do ambiente

3.2. Terminologia e Definições

Para que seja possível a representação de regras através de *Expressões de Contexto*, foi sugerida a seguinte metodologia e definições:

Nome (N): *Trata-se do substantivo que nomeia algum elemento que se encontra no ambiente, por exemplo: local, hora, data, contador de acesso e assim por diante.*

Operador (P): *É um operador lógico do conjunto {>,<,≥ ,≤,≠,=}. Esse conjunto pode ser estendido por parte do usuário. Por exemplo, pode ser definido um operador IN para retratar o fato de um instante de tempo estar dentro de um intervalo.*

Valor (V): *É o valor coletado do ambiente naquele momento relativo ao elemento que está sendo levado em consideração, esse valor é variável, por exemplo, o elemento pode ser á hora e seu valor 8am.*

Definição 1 (Propriedade de Contexto P)*: Uma propriedade de contexto é uma característica do contexto naquele momento, os atributos que o descrevem, e pode estar associada a quem executa ou sofre a ação. A propriedade de contexto é representada pela tupla <N,P,V> e assim definida:*

$$P = (N,P,V)$$

Definição 2 (Entidade de Contexto E): *Elemento do ambiente que sofre ou executa uma ação. Uma entidade de contexto possui um conjunto de propriedades de contexto a ela associadas as quais descrevem as características do ambiente ao seu redor no momento da requisição de acesso. Sendo assim* $\forall P \in E$*, Logo*

$$E=\{P_1,P_2,...,P_n\}$$

n *= Indica a possibilidade de um número variado de propriedades associadas a uma Entidade.*

Exemplos de Entidade de Contexto: ***Sujeito e Objeto****.*

Definição 3 (Expressão de contexto EC): *Descreve determinada situação que mapeia um contexto. Pode ser representada por uma Entidade de Contexto e suas propriedades ou por um Conjunto de Entidades de Contexto com suas propriedades. Uma Expressão Contextual pode possuir qualquer Entidade de Contexto associada a ela. Sendo assim, pode-se defini-la como:*

$$EC=\{E_1,E_3,...,E_n\}$$

n*: Indica um número variado de Entidades em cada Expressão de Contexto.*

Então, pode-se dizer que $\forall E \in EC$***.*** *No exemplo citado na figura 6 há duas entidades de contexto, logo:* ***EC*** *= (E1,E2).*

EC pode ser traduzida como:

$EC = (E1(P1 \cap P2) \cup E2(P1)) =$

EC = (entidade1=*sujeito* (função=*chefe da UTI* ∩ local=*UTI)* ∪ (entidade2=*objeto* (local=*UTI)*).

O significado de expressão contextual leva a definição de *contexto:* "encadeamento de informações sobre um dado ambiente" ou "conjunto de idéias, situações, eventos e informações necessárias para o correto entendimento do ambiente" em que as informações são mostradas na forma de propriedade. Uma *expressão contextual*, então, pode ser reconhecida como "*um conjunto de propriedades relacionadas a uma ou mais entidades*", ou seja, contém informações que caracterizam as entidades. Assim, entende-se um *contexto* como *um encadeamento de propriedades relacionadas a entidades em um ambiente***,** o que pode ser traduzido como um *Conjunto de Expressões Contextuais.* Sendo assim, chega-se a definição de contexto a seguir:

Definição 4 (Contexto CTX): *Características do ambiente formadas por uma ou mais Expressão de Contexto.*

$$EC_1 \times EC_2 \times \ldots \times EC_N \rightarrow CTX.$$

O conjunto de Expressões de Contexto relacionadas determinam um contexto. No exemplo da figura 6 é possível construir uma Expressão de Contexto para cada Entidade ou apenas uma envolvendo as duas Entidades, portanto, $EC_1 \rightarrow CTX$.

3.3. CABEC como extensão do RBAC

O CABEC leva em conta *Expressão de Contexto* para possibilitar nas autorizações de acesso a verificação dessas expressões. No RBAC uma autorização é expressa pelo conjunto <***p,obj,opr***>, onde ***p*** corresponde ao perfil para o qual um privilégio é estabelecido, ***obj*** é o recurso ou objeto para qual o privilégio se aplica, ***opr*** é o tipo de privilégio requisitado. Para integrar as expressões contextuais no CABEC deve-se estender esse conjunto permitindo a introdução do elemento contexto ***ctx***. A avaliação de um contexto ***ctx*** tem quatro respostas possíveis: *negado, permitido, indefinido e indeterminado.* O conjunto anterior pode então ser reescrito da seguinte forma:

Autorização: <***p,obj,ctx,opr***>

Onde ***ctx*** pode ser mapeado por uma política de autorização de acesso através das expressões de contexto na forma de propriedades de contexto. Quando ocorre a interação do usuário com o sistema o contexto é mapeado pelo CABEC. Esse contexto será comparado com os vários contextos que são cadastrados previamente em forma de Políticas de Controle de Acesso (PCAs); se um contexto corresponder com algum contexto cadastrado em alguma política o acesso poderá ser concedido.

3.4. Construindo regras de acesso através de Expressões de Contexto

No modelo CABEC uma regra é representada por uma *Expressão de Contexto* e uma política por um *Contexto* conforme pode ser observado na figura 07. Com isso é possível agregar várias regras dentro da mesma política ou construir várias políticas com regras separadas.

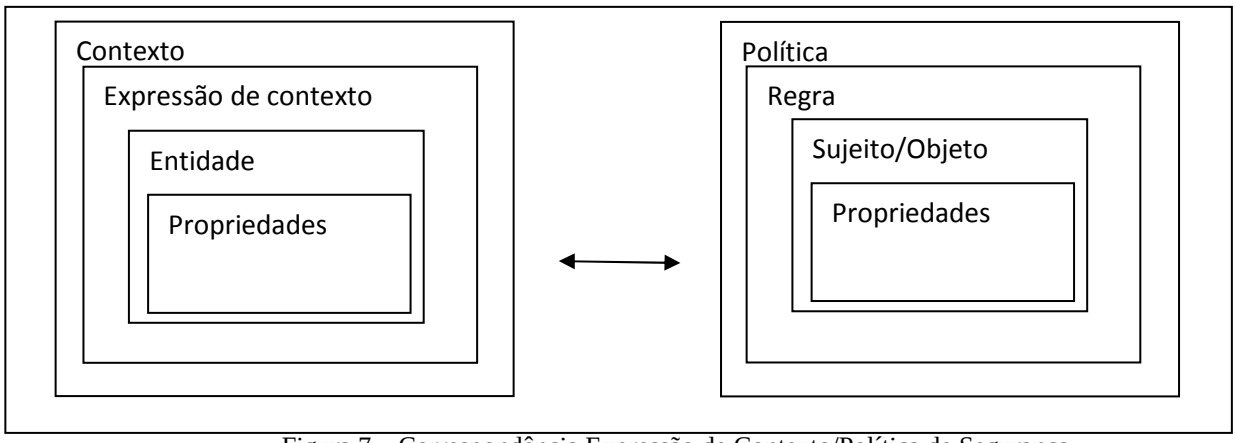

Figura 7 – Correspondência Expressão de Contexto/Política de Segurança

Uma Política de Controle de Acesso (PCA) pode ser representada como um contexto predefinido: PCA=CTX, sendo que Expressões de Contexto determinam um contexto (ECs→CTX), então basta verificar se as *Expressões de Contexto* retiradas do ambiente no qual ocorre o acesso se refletem em alguma PCA existente. Na figura 08 é feito o mapeamento de regras de acesso dentro da definição de contexto proposta pelo CABEC. São mostradas duas

regras de acesso em uma política de segurança para o domínio de saúde. A primeira $EC_1=(E_1(P_1 \cap P_2))$ pode ser traduzida como:

$EC_1=(Sujeito((tempo,>=,10:00) \cap (função,=,enfermeira)$

A segunda, $EC_2=(E_2(P_1 \cap P_2))$ pode ser traduzida por,

$EC_2=(objeto((contador,=,2) \cap (local,=,UTI)$

O contexto informado agrega duas *Expressões de Contexto* do domínio da saúde, uma para a entidade sujeito e outra para a entidade objeto, e dentre elas uma precisa ser correspondida para que o acesso seja concedido. Na primeira *Expressão de Contexto* EC_1 um *Sujeito (Entidade)* na *Função* de *Enfermeira (Propriedade)* pode acessar determinado recurso das *10:00 horas (Propriedade)* em diante, na segunda EC_2 o *objeto (Entidade)* pode ser acessado do *local UTI (Propriedade)* com um *limitador de dois (Propriedade)* usuários simultâneos. Os parâmetros levados em consideração para concessão do acesso são dinâmicos (hora, função, local, limitador) seus valores podem variar de requisição uma requisição para outra. A flexibilidade é caracterizada pela possibilidade de combinação das *Expressões de Contexto*, cada expressão é uma regra positiva, ou seja, capaz de conceder o acesso. No exemplo são duas possibilidades, essa quantidade é determinada pela política de segurança construída de modo específico em cada caso. Outra característica da flexibilidade é a quantidade de Entidades e Propriedades consideradas na construção da política, no exemplo são consideradas duas entidades para cada expressão de contexto e a cada uma delas são associadas duas propriedades.

A PCA da figura 9 admite o acesso a determinado objeto no domínio de administração de uma rede, o *objeto (Entidade)* é o arquivo *httpd.conf* que é responsável pelas configurações do servidor Web Apache, a regra diz que ele pode ser acessado através da *rede sem fio (Propriedade)*, desde que o usuário do sistema esteja na *função de Administrador da Rede* (*Propriedade)*.

```
<Context>
 <Express_Context>
   <Entity="Sujeito">
     <Property Name="TEMPO">
```

```
            <Operador OP=">="/>
             <Value V="10:00"/>
          </Property>
          <Property Name ="Funcao">
             <Operador="="/>
             <Value="Enfermeira"/>
          </Property>
        </Entity_Type>
      </Express_Context>
      <Express_Context>
       <Entity="Objeto">
          <Property Name="contador">
            <Operador OP="="/>
            <Value V="2"/>
          </Property>
          <Property Name ="local">
             <Operador="="/>
             <Value="UTI"/>
          </Property>
       </Entity>
      </Express_Context>
</context>
```

Figura 8 - PCA domínio saúde.

Através das credenciais da requisição o CABEC monta o contexto e identifica o usuário (*Entidade)* com *Perfil Gerente de Informática (Propriedade)* tentando acessar o *objeto httpd.conf (Entidade)* para *leitura* de uma *rede sem fio (Propriedade).* As credenciais tem o formato: <***p,obj,ctx,opr***> traduzindo **<*Gerente de Informática, httpd.conf, CTX, leitura*>.** Onde CTX é montado pelo CABEC, trata-se do contexto extraído do ambiente, então:

$CTX=(EC_1)$ e $EC_1=(E_1(P_1) \cap E_2(P_1))$ onde:

EC_1 = ***(sujeito(função=gerente de informática) ∩ (Objeto(local=Rede Móvel))***

Esse contexto é encaminhado para que possa ser avaliado em relação à PCA correspondente. A avaliação é a comparação entre o valor das propriedades extraído do ambiente com o valor das propriedades constantes nas políticas. O resultado será disponibilizado dentro das quatro opções de resposta (*negado, permitido, indefinido e indeterminado).* Nesse caso será *negado* tendo em vista que a política somente permite o acesso através da rede móvel se o papel ou perfil do sujeito é ***Administrador de Rede***.

```
<Context>
 <Express_Context>
  <Entity_Type="Sujeito">
    <Property Name="Função">
     <Operador OP="="/>
     <Value V="Administrador da Rede"/>
    </Property>
     <Property Name="local">
       <Operador OP="="/>
       <Value V="Rede Movel"/>
     </Property>
    </Entity_Type>
    <Entity_Type="Objeto">
     <Property Name="nome_objeto">
      <Operador OP="="/>
      <Value V="httpd.conf"/>
     </Property>
</Entity_Type>
</Express_Context>
</context>
```

Figura 9 - PCA para domínio redes de computadores.

A regra da figura 09 possui duas entidades aninhadas em uma única expressão de contexto, a primeira *(sujeito)* possui duas propriedades associadas e a segunda *(objeto)*

somente uma propriedade. As propriedades *função* e *local* são dinâmicas, pois o usuário pode exercer funções diferentes em momentos diferentes, estar na rede cabeada em um momento e na rede móvel em outro. A possibilidade de combinar várias entidades para montar uma expressão de contexto reflete a flexibilidade do modelo, uma vez que, o nível de detalhe considerado é decidido por quem monta a política de segurança. Se fosse necessário considerar mais uma *Entidade*, por exemplo, *subsistema* do qual o usuário está fazendo acesso, tal característica pode ser representada conforme demonstrado na figura 10.

```
<Entity_Type="subSistema">
    <Property Name="nome">
     <Operador OP="="/>
     <Value V="Sistema_monitoramento_distancia"/>
  </Property>

<Entity/>
```

Figura 10 - Entidade de contexto referente a um subsistema

O nome do subsistema considerado reflete um módulo que pode ser utilizado especialmente da rede externa, tal módulo pode conter mecanismos de criptografia para tráfego de mensagens e com isso ser mais seguro.

Na figura 11 demonstra-se uma política de acesso para o domínio de imagens geoespaciais com alta resolução. Nela um *Perfil "Defesa Civil" (Propriedade)* pode acessar uma *imagem (Entidade)* de *resolução de 2 metros (Propriedade)* mesmo estando em uma *rede sem fio (Propriedade)*. Através da credencial de autorização ***<defesa_civil, img_satelite, CTX, leitura>*** pode-se montar a *Expressão de Contexto* segundo os elementos do CABEC:

$CTX=(EC_1)$ e

$EC_1=((E_1(P_1 \cap P_2)) \cap (E_2(P_1 \cap P_2)))$ onde:

EC_1 = ***((sujeito(Função=DefesaCivil)*** $\cap$ ***(local=RedeMovel))*** $\cap$ ***(Objeto(nome_objeto=imgSatelite)*** $\cap$ ***(Resolução=2M)))***

```
<Context>
 <Express_Context>
  <Entity_Type="Sujeito">
   <Property Name="Função">
    <Operador OP="="/>
    <Value V="DefesaCivil"/>
   </Property>
    <Property Name="local">
     <Operador OP="="/>
     <Value V="Rede Movel"/>
   </Property>
  </Entity_Type>
  <Entity_Type="Objeto">
   <Property Name="nome_objeto">
    <Operador OP="="/>
    <Value V="imgSatelite"/>
   </Property>
   <Property Name="resolucao">
    <Operador OP="="/>
    <Value V="2M"/>
   </Property>
  </Entity_Type>
 </Express_Context>
</context>
```

Figura 11 - PCA para domínio dados geoespaciais.

Nessa *Expressão de Contexto* duas *Entidades* são consideradas *(Sujeito e Objeto)* e para cada entidade são consideradas duas propriedades. A quantidade de propriedades pode variar, essa possibilidade demonstra a flexibilidade que o CABEC permite em relação às características do ambiente que são consideradas relevantes por quem monta as políticas.

Na figura 12 é demonstrada a construção de uma política de segurança para o domínio de uma residência. Nela *uma expressão de contexto* é formada por uma *Entidade* sujeito. Para

essa entidade são associadas três propriedades. A *categoria, local e horário.* O horário é chamado de *horario_depois_licao* o que reflete um intervalo de tempo previamente cadastrado no qual a *categoria criança* pode assistir TV. Então, com uma credencial ***<usr_id, aparelho_tv, CTX, ligar*** > o CABEC monta o contexto e avalia se ele está de acordo com o que a regra representa. Nesse caso, se *usr_id* é da categoria *criança*, está acessando da *sala_de_estar* e no intervalo *horário_depois_licao* a ação *ligar* será autorizada para *aparelho-tv*.

```
<Context>
 <Express_Context>
  <Entity_Type="Sujeito">
   <Property Name="categoria">
    <Operador OP="="/>
    <Value V="Crianca"/>
   </Property>
    <Property Name="local">
     <Operador OP="="/>
     <Value V="Sala de estar"/>
   </Property>
    <Property Name="horario">
     <Operador OP="="/>
     <Value V="Horario_dapois_licao"/>
   </Property>
  </Entity>
 </Express_Context>
</context>
```

Figura 12 - PCA para domínio de uma residência

Essa política demonstra uma possibilidade de realização da ação *ligar* para o objeto *aparelho_tv*, porém, podem ser construídas outras políticas para esse caso. Por exemplo, pode ser construída uma política com uma *Expressão de Contexto* que contém a *Entidade Sujeito*

associada às propriedades *categoria Empregada, local cozinha, horário novela das 6.* O administrador do CABEC pode criar essa política ou simplesmente combiná-la com uma já especificada, no caso a da figura 12, o que remeteria a duas possibilidades (contextos) de acesso definidas na mesma política, isso demonstra a flexibilidade permitida pelo modelo em relação à quantidade de regras nas políticas que fica a critério do administrador do CABEC. O mesmo irá definir como vai organizar as políticas de acordo com o que considerar mais interessante, pode separar as regras de acesso em várias políticas ou deixá-las combinadas em uma única.

```
Receber requisição de acesso
Montar Expressão de Contexto
Selecionar todos os contextos que satisfaçam a requisição de acesso:<P,O,OPR>
    flage = false
    Enquanto houver contexto  ou  flagE == false
    inicio
            Obter Expressões de Contexto do contexto
            Enquanto houver Expressões de Contexto ou flagE == false
            Inicio
                flagP = True
                enquanto encontrar Entidade de Contexto
                 inicio
                            Para cada Propriedade
                            Se Propriedade == falsa
                           flagP = false
              fim
      Se flagP == true
      flagE = true
   fim
fim
```

Figura 13. Algoritmo de avaliação de Contexto.

3.5. Algoritmo de avaliação de contextos

Uma vez que é possível a representação de contexto, tanto na composição da requisição do usuário, quanto nas políticas de segurança, faz-se necessário um algoritmo que faça a avaliação do contexto extraído do ambiente com o contexto mapeado pela política, para identificar se os dois são correspondentes ou não. Para tanto, foi proposto o algoritmo da figura 13.

O algoritmo proposto recebe a requisição de acesso na forma de predicado e recupera os contextos que se referem ao *Objeto O, Papel P e Modo de Acesso OPR.* Cada contexto tem suas *Expressões de Contexto,* é necessário que as mesmas sejam avaliadas para conferir se existe alguma que corresponde ao contexto extraído do ambiente.

Cada contexto representa uma política de segurança, e cada política possui suas regras. As regras definem um meio de acessar determinado objeto ou recurso computacional para execução de alguma ação. No algoritmo as políticas são consideradas contextos, e suas regras expressões de contexto, de modo que, em uma política pode existir mais de uma forma de acesso possível. Para cada expressão de contexto são verificadas as *Entidades* e suas *Propriedades*, quando todas as propriedades são verdadeiras a expressão de contexto extraída do ambiente é válida e o acesso concedido.

Mais de uma política pode ser construída para cada tipo de acesso, sendo assim, podem coexistir políticas de segurança com contextos diferentes para conceder o acesso ao mesmo objeto para aquela mesma operação. Porém, o algoritmo a partir do momento que encontra uma Expressão de Contexto que reflita a Expressão de Contexto montada do ambiente não verifica mais as outras políticas, por considerar que já foi encontrada uma regra positiva que concede o acesso.

As variáveis flage e flagp são ambas para controlar os testes de Expressão de Contexto verdadeira e Propriedades da Entidade verdadeiras. É preciso testar todas as propriedades informadas do ambiente em cada Expressão de Contexto para conferir se possuem os mesmos valores das propriedades pré-cadastradas nas políticas, uma vez confirmadas a Expressão de Contexto é considerada válida e o algoritmo já pode retornar o estado *permitido*. Caso não sejam encontradas políticas para a credencial <P,OBJ ,POR> o algoritmo retornará o estado *não aplicável.* Se forem encontradas políticas, mas nenhuma Expressão de Contexto com

todas as propriedades verdadeiras o retorno será *negado.* Caso ocorra algum erro no processo de avaliação, seja ele lógico ou de sintaxe o retorno é *indeterminado.*

3.6. Conclusões do capítulo

Considerando que existem modelos de controle de acesso baseados em contexto que não demonstraram a construção de políticas flexíveis para diferentes ambientes, o CABEC através das demonstrações da sessão 3.4 revelou que tanto no ambiente da saúde como no de administração de uma rede, dados geoespaciais e ambientes residenciais é possível a construção de regras pela combinação das propriedades contextuais, essas regras são parâmetros das políticas de segurança de uma instituição. Os elementos definidos para o modelo permitem abstrair dados de ambientes diferentes através das Entidades de Contexto e suas Propriedades. Cada Entidade pode considerar inúmeras propriedades do ambiente que a cerca, isso demonstra a abrangência relacionada com os detalhes envolvidos na construção de políticas de segurança, aspecto a ser considerado no atual cenário onde é possível conectividade com mobilidade.

O algoritmo que implementa o modelo dá ênfase na otimização do processamento, uma vez que quando encontra uma regra positiva já retorna um estado permissivo para o cliente, não avaliando outras regras que também permitem o acesso para aquele recurso naquela ação.

4. IMPLEMENTAÇÃO

Este capítulo mostra como foi implementado o modelo CABEC, justificando a escolha das tecnologias e arquitetura proposta. As principais tecnologias escolhidas para implementação do CABEC foram Web Services (WEB SERVICES, 2009) e XACML (OASIS XAML, 2009) as quais são fundamentadas a seguir.

4.1. Web Services

Web Services são aplicações modulares independentes de linguagem de programação ou sistema operacional. Podem ser descritos através de uma linguagem de descrição de serviços, são publicados em um registro de serviços, descobertos através de mecanismos de busca e invocados por sua *Application Program Interface* – API. Também podem ser combinados entre si para compor uma nova funcionalidade.

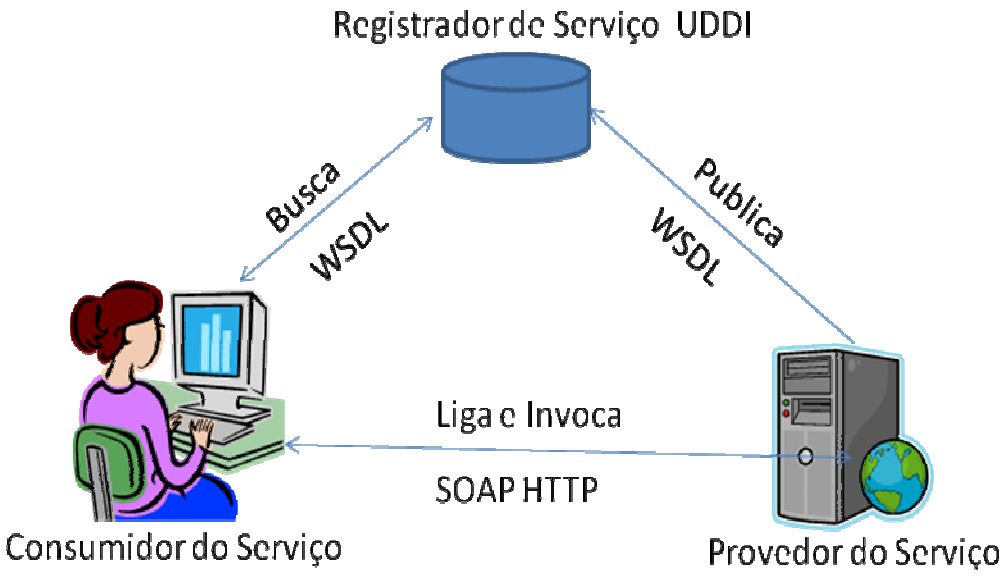

Figura 14 - Estrutura de funcionamento de um *Web Service*.

A estrutura de funcionamento de um *web service* pode ser vista na figura 14, nela pode-se notar um provedor de serviço, um registrador de serviço e um consumidor que

precisa consultar o registrador para saber sobre a existência e descrição de determinado serviço para conseguir utilizá-lo ligando-se a ele e invocando-o.

Um *Web Service* é uma caixa preta, onde o cliente que consome o serviço tem acesso apenas à funcionalidade disponibilizada, cabendo a ele saber como empregar tal serviço. Isso é proporcionado pela arquitetura modular auto-contida do *Web Service* que disponibiliza apenas sua API para ser acessada pela rede como única forma de iteração com sistema que possui um serviço publicado. Esta API suporta a comunicação programa para programa, o que permite que aplicações se comuniquem usando XML através da própria Web (MARTINS, 2007).

A aceitação em larga escala do *Web Service* acarretou na criação de tecnologias agregadas, dentre elas cabe destacar a Linguagem de Descrição de *Web Service*, *Web Services Description Language* - WSDL, que descreve a interface e o protocolo de comunicação, e o *Simple Object Access Protocol* - SOAP, protocolo baseado em XML para Chamada de Procedimento Remoto *Remote Process Call* RPC.

As arquiteturas tradicionais de sistemas distribuídos como *Distribuited Common Object Model* - DCOM e *Commom Object Request Broker Architecture* CORBA apresentam alta sensibilidade a mudanças devido ao alto grau de acoplamento e interdependência entre as partes que compõem os modelos. O padrão *Web Service* visa proporcionar ganhos de estabilidade onde às arquiteturas tradicionais deixam a desejar com a adoção de padrões abertos codificados em XML para realizar a comunicação dos componentes que formam esse sistema distribuído e fornecendo uma arquitetura orientada a serviço para comunicação entre seus componentes (NETO, 2004).

A utilização crescente de *Web Service* é impulsionada por vários fatores, um deles, de maior relevância, é o fato de que é baseado em XML, que facilita tanto a integração de sistemas do mesmo domínio quanto à integração de sistemas que desejam compartilhar funcionalidades ou serviços com domínios parceiros de forma fácil e barata. Mesmo ainda apresentando alguns pontos e partes que necessitam de amadurecimento em relação a aspectos de segurança na troca de informação, esforços vêm sendo desenvolvidos para sanar essas deficiências com o uso do *WS Security* (HANAUER, 2004).

WS-Security suporta, integra e unifica vários modelos, mecanismos e tecnologias de segurança em uso no mercado, permitindo que vários sistemas possam inter-operar em plataformas e linguagens neutras. As novas especificações de segurança definem um conjunto de padrões para extensões SOAP ou para cabeçalhos de mensagens, utilizados para oferecer

maior integridade, confidencialidade e autenticação das mensagens com o sistema de transmissão de mensagens SOAP (WATHIER, 2005).

A especificação "*Web Services Security: SOAP Message Security 1.0*", pelo grupo OASIS (*Organization for the Advancement of Structured Information Standards*) descreve mecanismos de segurança na troca de mensagem SOAP ao definir um perfil no uso da Assinatura XML (XML *Signature*) e da Criptografia XML (XML *Encryption*) para garantir integridade e confidencialidade para mensagens SOAP (OASIS, 2009).

4.2. XACML

O *eXtensible Access Control Markup Language* (XACML) é uma linguagem de marcação baseada em XML, utilizada em ambientes onde se deseja representar políticas que controlem o acesso de usuários a recursos do sistema (ex.: um arquivo, uma imagem, ou informações relevantes armazenadas em banco de dados) (HIGASHIYAMA, 2005).

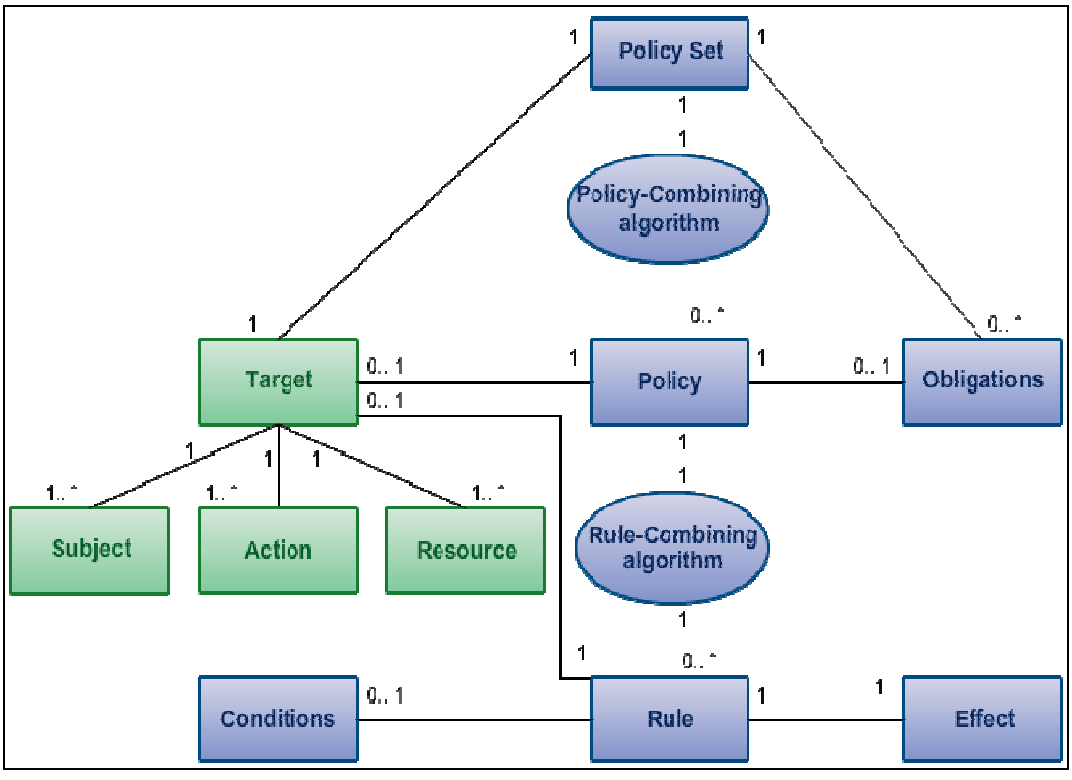

Figura 15 - Estrutura da política XACML

Uma política XACML possui como elementos fundamentais os atributos *Policy, Rule*, e *Policy Set.* Conforme demonstrado na figura 15. *Policy* é o atributo base que a política XACML deve conter para ser avaliada, os principais componentes da *Policy* são *Target, Rule, Obligations* e o atributo "*Rule-Combining algorithm*". *Rule* é um elemento que contém expressões booleanas que podem ser avaliadas isoladamente que possibilitam as políticas definir regras isoladas que serão utilizadas na avaliação de uma *authorization decision*. *Target* contém o cabeçalho da política, no qual é possível a especificação dos dados relacionados ao *sujeito, objeto* e *ação* sobre os quais a regra ou conjunto de regras serão aplicados. *Obligations* são diretrizes das políticas aplicáveis no Ponto de Execução da política. Podem ser transmitidas em forma de orientação para a aplicação, por exemplo, acesso de *alteração* depois das *12h00min horas.*

O elemento *PolicySet* contém um conjunto de elementos *Policy* ou outros elementos *PolicySet* e um algoritmo especifico para combinar os resultados das suas avaliações. Os elementos *Subject, Resource* e *Action* ilustrados na figura 11 representam os atributos que devem ser especificados em uma requisição XACML referente ao sujeito que está requisitando o acesso ao recurso e que ação pretende realizar. Esses atributos possuem ligação com o atributo *Policy* através do elemento *Target.*

A combinação de algoritmos representados pelas classes *Policy-Combining algorithm* associado com *Policy* e *Rule-Combining algorithm* relacionado com *Rule,* podem ser *Deny-overrides, Permit-overrides, First-applicable* ou *Only-one-applicable.* A escolha de um deles pelo gerente de segurança determina o modo como as regras serão avaliadas:

Deny-overrides: Se em uma avaliação de um conjunto de regras o resultado é uma negação "*Deny*", esta tem precedência sobre qualquer outro resultado.

Permit-overrides: se em uma avaliação de um conjunto de regras o resultado é uma permissão "*Permit*", esta tem precedência sobre qualquer outro resultado.

First-applicable: As avaliações das regras seguem a seqüência que está descrita na *Policy*. Se em uma avaliação de um conjunto de regras, a primeira regra pode ser aplicada, prevalece seu resultado, podendo ser uma permissão "*Permit*" ou uma negação "*Deny*".

Only-one-applicable: Se em uma avaliação de um *PolicySet*, possa ser aplicada uma *Policy*, prevalece seu resultado, caso contrário pode retornar "*Indeterminate*" ou "*NotApplicable*" se não for aplicada a nenhuma das políticas da *PolicySet*.

Um dos objetivos do XACML é definir um padrão descritivo para políticas de acesso contendo informações sobre o usuário, recurso e ação. Para alcançar esse objetivo propõe uma arquitetura com as entidades *Policy Enforcement Point* – PEP, *Policy Decision Point* – PDP e

Administration Point – PAP. PEP representa o ponto de execução da requisição de acesso, nele são incorporados os atributos do requisitante, do objeto e da ação. PDP é o ponto de decisão, realiza a análise da requisição com o confronto desta com as políticas de acesso. PAP são as políticas de controle de acesso definidas pelo administrador do sistema (LIMA, 2007). A figura 16 ilustra o fluxo de informações das entidades que usam XACML. Observe que um evento externo encaminha ao PEP uma requisição de acesso para ser processada pelo *context handler*, também chamado de controlador de contexto. O PEP recebe os dados do evento externo, podendo ou não agregar informações que melhor representam o contexto da requisição externa e cria uma *request* para o controlador de contexto.

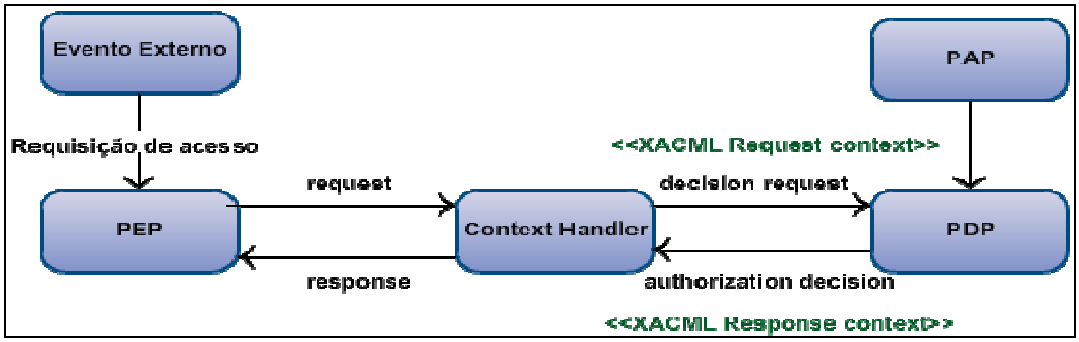

Figura 16 - Fluxo de dados XACML

O *context handler* forma uma requisição chamada *decision request* codificada em XACML com a informação recebida do PEP e disponibiliza ao PDP para decidir se a solicitação será concedida ou negada com base nas políticas armazenadas no PAP. A decisão do PDP chamada de *authorization decision* é encaminhada ao *context handler* que converte para o formato nativo do evento externo e disponibiliza a resposta ao PEP através da *response* que por sua vez informa o resultado da decisão ao evento externo.

Para motivar a criação de políticas em XACML sem que o gerente de segurança necessite dominar toda complexidade de sua estruturação, uma alternativa é o Sistema de Gerenciamento de Políticas de Controle de Acesso SGPCA (LIMA, 2007). O Sistema possui interface gráfica afastando do usuário a complexidade relacionada à estruturação das políticas na sua criação, além disso, faz detecção de conflitos em tempo de criação das políticas. A restrição de tal sistema consiste em permitir apenas criação de políticas que levam em conta propriedades temporais ou de localização.

4.3. Arquitetura CABEC

Visando maior facilidade de manutenção e aprimoramento do código do serviço CABEC, foi adotada uma arquitetura de software dividida em níveis ou camadas, como demonstra a figura 17.

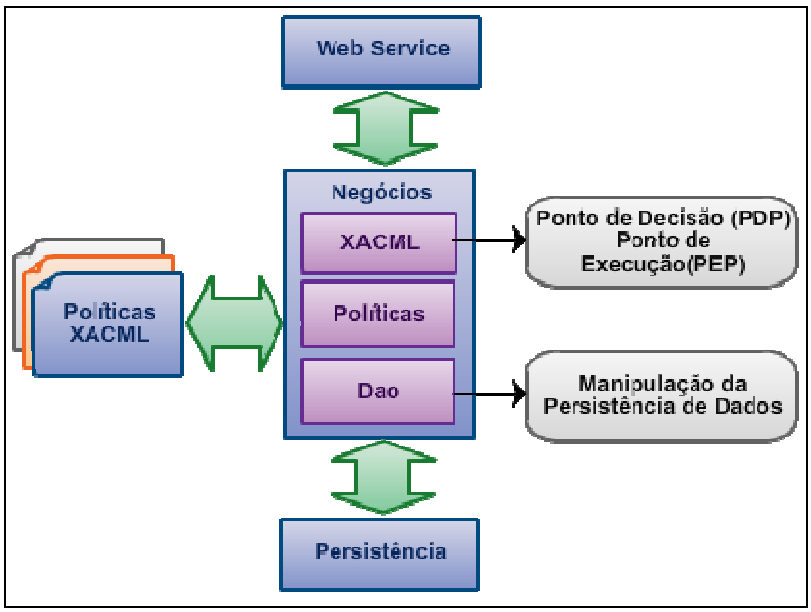

Figura 17 – Arquitetura do Cabec

Web Service: Camada responsável por tornar o sistema adaptável a diferentes linguagens de programação. Disponibiliza o endereço WSDL do serviço CABEC após a publicação do mesmo. Assim é possível que qualquer linguagem de programação consuma o serviço. Para transportar os dados de requisição e resposta na rede é utilizado o protocolo SOAP em conjunto com o protocolo HTTP.

Negócios: A camada possui três módulos: *Dao, Politicas e XACML*. O módulo *Dao* possui métodos para manipular os dados armazenados na camada de persistência. Em *XACML* estão os métodos responsáveis por montar as propriedades das Entidades de Contexto envolvidas na requisição obtendo dados da camada de persistência. De posse desses dados a requisição XACML é formada e confrontada com as políticas XACML obtidas do módulo de Políticas

no Ponto de Decisão com o objetivo de encontrar as políticas que sejam aplicadas as credenciais da autorização e avaliar as mesmas para formar a resposta da requisição.

Persistência: Contém o mapeamento das entidades do banco de dados escolhido para objetos da linguagem de programação Java através da configuração da classe responsável pela integração do banco de dados adotado com o Hibernate. O Hibernate é uma solução open-source para Mapeamento Objeto/Relacional *Object/Relational Mapping* - ORM. ORM é um procedimento que consiste em mapear entidades de diferentes SGBD para objetos Java, seu uso torna a aplicação mais flexível em relação ao banco de dados utilizado permitindo ao desenvolvedor a migração entre diferentes tecnologias de base de dados.

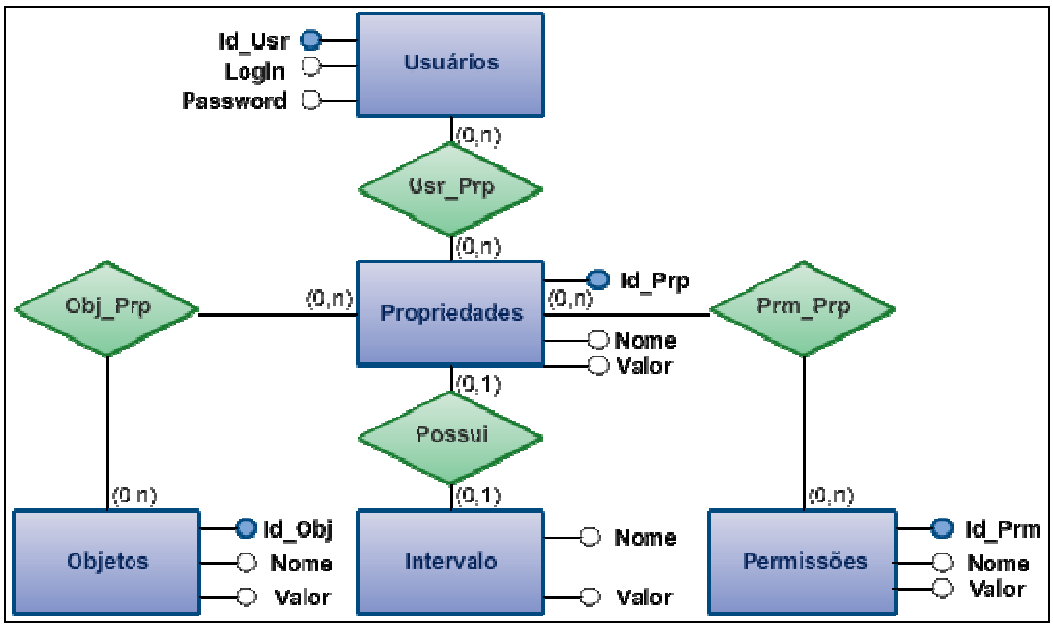

Figura 18 – Modelo de Dados CABEC

As informações do banco de dados são utilizadas na construção de expressões contextuais que formam a requisição de acesso no ponto de execução da política. O modelo de dados do CABEC é mostrado na figura 18. Nele é possível perceber duas Entidades de Contexto: Usuários e Objetos, ambas com um conjunto de propriedades. Isso permite a construção de expressões de contexto. Permissões é uma tabela que permite ao modelo funcionar como o RBAC, caso seja do interesse do gerente de segurança ele pode simplesmente começar a considerar essas permissões e não precisará consultar as políticas em XACML para tornar o acesso somente dependente do perfil do usuário.

4.4. Sistema Cabec

Definida a estrutura do sistema e suas tecnologias de implementação foi construído o diagrama de classes do sistema para especificar como o modelo deveria ser construído com base na arquitetura proposta, conforme demonstrado na figura 19.

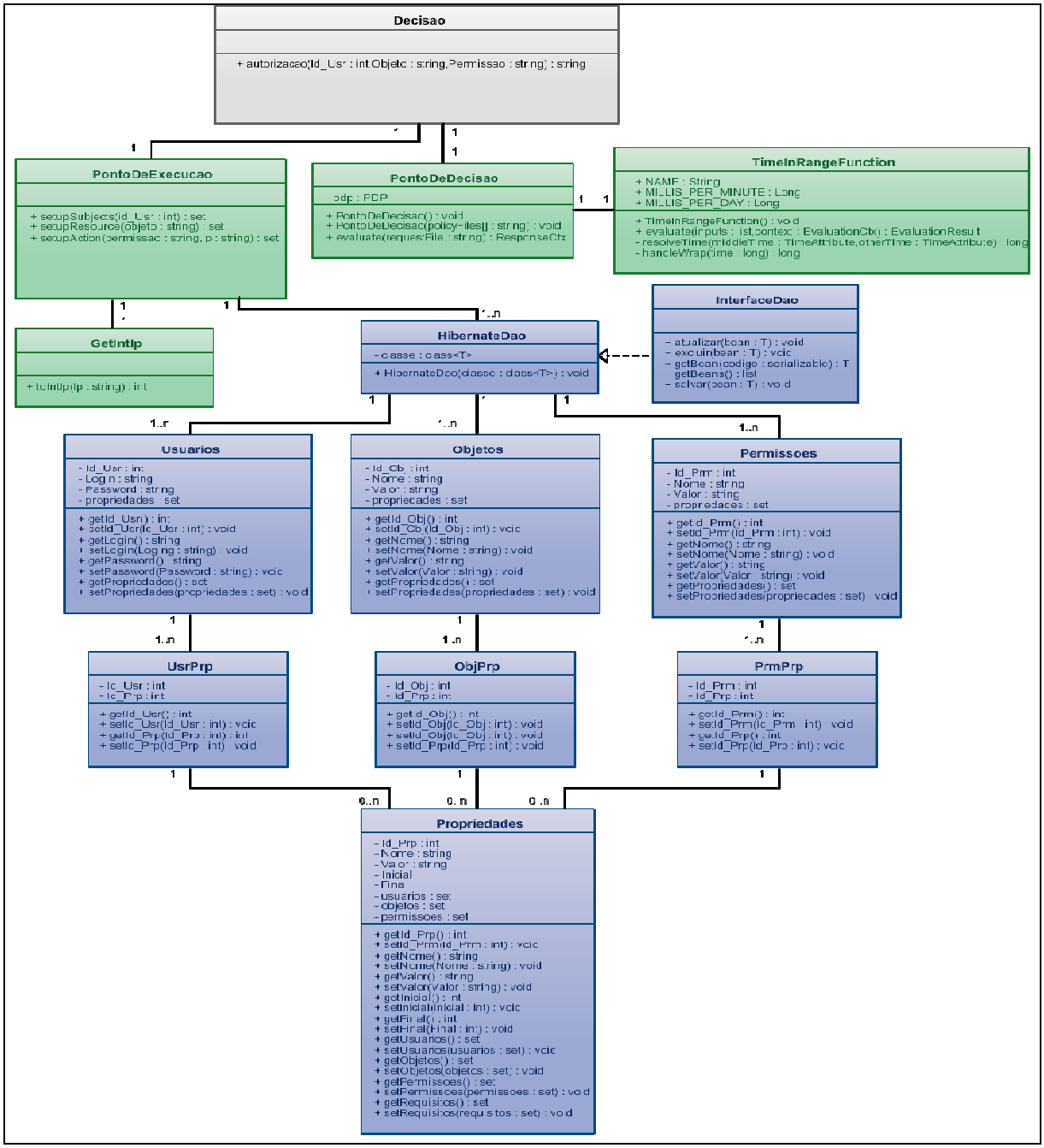

Figura 19 - Diagrama de Classes do CABEC

Na figura 20 é possível ver como ficou a estrutura de implementação do serviço CABEC.

Figura 20 – Implementação das Classes do CABEC

A figura 21 ilustra o funcionamento interno do sistema usando as classes especificadas no diagrama de classes e as camadas definidas na arquitetura do sistema seguido da explicação da ação realizada no nível utilizado.

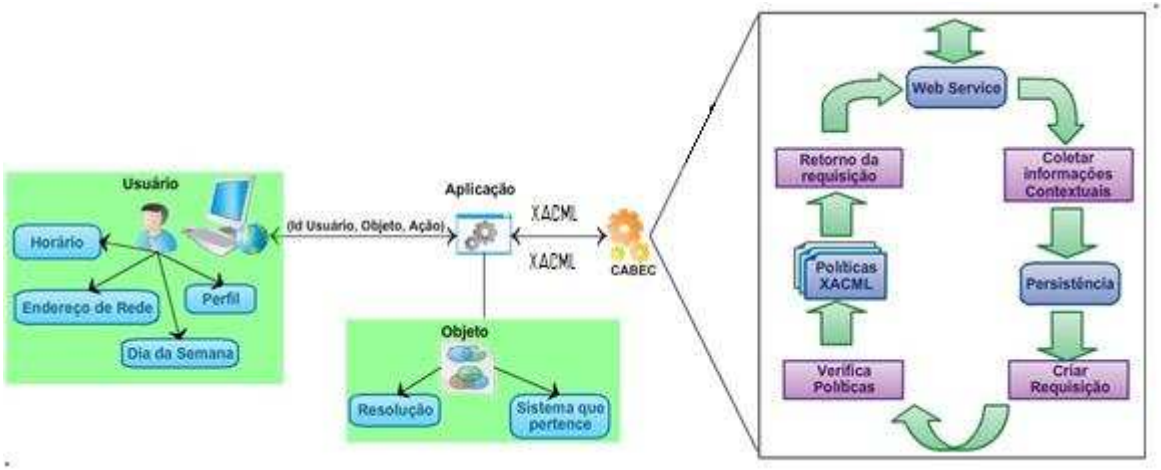

Figura 21 - Funcionamento durante uma requisição de acesso.

O usuário autentica-se no sistema cliente informando seus dados, o sistema por sua vez, no momento que for requisitado o acesso a um objeto envia para o serviço CABEC: Id do Usuário, Objeto a acessar e Ação a realizar. O serviço recebe as credenciais do usuário, monta a *Context Request* através da coleta de informações contextuais da camada de persistência e encaminha para o módulo de políticas onde ocorre a conferencia. Uma vez feita a conferencia o serviço CABEC monta o retorno da requisição no formato XACML *Context Response*.

4.5. Conclusões do Capítulo

O CABEC foi implementado com uma arquitetura modular, isso facilita a manutenção do sistema. É um sistema de baixo acoplamento, pois utiliza tecnologias abertas, *Web Service* para comunicação e transporte dos dados, XACML para representação e Hibernate para mapeamento das classes em tabelas. A linguagem Java na qual foi construído é portável para diferentes sistemas operacionais o que também demonstra a flexibilidade para sua instalação e publicação.

5. DISCUSSÃO E VALIDAÇÃO

O modelo CABEC utiliza contexto para controlar o acesso a recursos, outros modelos já fazem isso, porém, de forma diferente em relação a sua representação e aplicação, sendo assim esse capítulo faz uma comparação entre os modelos baseados em contexto considerados mais relevantes e o modelo CABEC. Também, é relatado o estudo de caso no qual o CABEC foi integrado com o Banco de Dados de Geodesastres do Instituto Nacional de Pesquisas Espaciais (INPE) em Santa Maria para fins de testes.

5.1. Discussão

O modelo TRBAC (*Temporal Role-Based Access Control*) (BERTINO, et al., 2001), habilita perfis dentro de seus turnos, e seu aperfeiçoamento o GTRBAC (*Generalized Temporal Role-Based Access Control*) (JOSHI et al., 2003) consegue restringir a ativação dos perfis para os dias nos quais estão escalados para trabalhar. Essas funcionalidades são demonstradas através do registro no banco de dados de linhas nas quais é possível a representação de um evento periódico. O CABEC representa intervalos de tempo através das propriedades, assim como os intervalos de dias determinado funcionário está escalado para exercer sua função, sendo assim consegue exercer o mesmo controle que esse modelo e, além disso, permite a representação de outros contextos além dos temporais como demonstrado no capítulo III sessão 3.4.

O modelo E-RBAC (*Environment Role-Based Access Control)* foi desenvolvido por (COVINGTON et al.,2001) com o objetivo de ativar Papéis ambientais no controle de acesso especifico de residências, e o modelo XORBAC (*eXtended Object Role-Based Access Control*) foi desenvolvido por (NEUMANN, 2003) estende o RBAC básico com a associação de restrições contextuais aos papéis de usuário para garantir uma autorização de acesso. Ambos os modelos coletam informações de sensores nas quais se baseiam para mapeamento do contexto, sendo assim, esses modelos ficam dependentes da tecnologia de sensores utilizada para captar a informação. Não é relatado pelos autores o que aconteceria na ocorrência de falhas, isso pode ser um problema para implantação das políticas de segurança, pois a verificação de dados ficaria comprometida. Além disso, os autores do XORBAC

relatam a necessidade de realização de mais testes demonstrando o funcionamento do modelo em outros domínios.

No CABEC o contexto é retirado de informações como o relógio do próprio computador e propriedades que são cadastradas pelo gerente de segurança, ele pode alterar essas propriedades sempre que for necessário, portanto, a dependência de dispositivos externos do tipo sensores para montagem do contexto é diminuída e fica a cargo do sistema cliente optar pelo uso dessa tecnologia ou não.

Outro modelo que estende o RBAC com a utilização de contexto é o MACA (*Middleware* de Autenticação e Controle de Acesso) (MOTTA, 2003), nesse modelo são relatados tipos de contexto associados à área da saúde. As expressões lógicas utilizadas no modelo são o resultado da combinação dos tipos de contexto e suas variáveis contextuais. Tanto na definição dos tipos de contextos como nas regras de acesso percebe-se a utilização de funções e tipos que são utilizados em sua maioria somente dentro desse domínio. O autor não demonstra a utilização de seu modelo com outros tipos ou funções referentes a um ambiente fora da área proposta. Além disso, esse modelo foi implementado com *Commom Object Request Broker Architecture* CORBA tecnologia com alta sensibilidade a mudanças devido ao alto grau de acoplamento e interdependência entre as partes que compõem os modelos (NETO, 2004). O CABEC não possui suas definições de contexto vinculadas a um ambiente em especifico, isso ajuda o mapeamento de diferentes contextos, além disso, foi implementado com a tecnologia *Web Service*, um padrão aberto que possui baixo acoplamento.

O modelo GEO-RBAC proposto em (DAMIANI & BERTINO, 2006) é uma extensão do RBAC para mapeamento da posição do usuário no espaço, o modelo espacial adotado é compatível com o *Open GeoSpatial Consostium* OGC. Esse modelo é especifico para mapeamento de contexto espacial não abrangendo outros ambientes.

O CIBAC (*Contextual Information-Based Access Control*) é proposto por (SOARES, 2006), estende o RBAC básico com a introdução de informações contextuais na regra de autorização de acesso. Nesse modelo o autor defende que existe a possibilidade de criação de regras em ambientes diferentes, porém não demonstra nenhum exemplo de como isso aconteceria. Outra observação em relação ao CIBAC se refere ao fato de que seu algoritmo de avaliação de contexto introduz o elemento cláusula, o qual não foi definido na especificação do modelo revelando incoerência entre a especificação e o algoritmo de implementação. O CIBAC foi implementado com uma arquitetura na qual três serviços trocam mensagens entre

si, isso aumenta o *overhead* na rede e pode comprometer a desempenho do sistema, no CABEC só existe um serviço englobando todas as funcionalidades do sistema.

O modelo Context-aware RBAC (KULKARNI D. & TRIPATHI) é embutido em um framework para programação e projeto de aplicações sensíveis a contexto. A característica nova desse framework é o alto nível de abstração para especificar requerimentos de controle de acesso baseado em contexto, O modelo CA-RBAC tem sido implementado através de um framework em XML, algumas simulações de seu uso em aplicações de computação pervasiva são mencionadas, porém, não existe relato de um teste prático do modelo.

Com o objetivo de diferenciar melhor os modelos entre si e suas potencialidades foi construída a tabela 04 com algumas características de classificação dos mesmos:

Modelo	**Demonstra representação de regras em mais de um tipo de ambiente**	**Não depende de sensores para o mapeamento do contexto**	**Tecnologia de implementação de baixo acoplamento**
TRBAC/GTRBAC		X	
ERBAC/XORBAC			
MACA		X	X
GSAM	X	X	
GEO-RBAC		X	
CIBAC		X	X
CA-RBAC	X		X
CABEC	X	X	X

Tabela 04 - Comparação entre modelos que usam contexto.

É importante à demonstração de criação de regras para diferentes tipos de ambiente (geoespacial, saude, temporal,...), pois isso revela maior abrangência em relação à utilização do modelo. A dependência de sensores pode comprometer o funcionamento do modelo caso não existam alternativas de tolerância a falhas e ao mesmo tempo compromete o cliente a implantar tal estrutura. Construir o modelo com tecnologias de baixo acoplamento aumenta a possibilidade de integração e uso.

5.2. Integração do CABEC com o Banco de dados de Geodesastres

O Instituto Nacional de Pesquisas Espaciais – INPE, implantou em seu Centro Regional Sul – CRS, o Núcleo de Pesquisa e Aplicação de Geotecnologias em Desastres Naturais e Eventos Extremos, denominado “Geodesastres – Sul”. Este Grupo de pesquisa busca atuar juntamente com órgãos como a defesa civil, corpo de bombeiros, brigada militar e outros, numa parceria para a prevenção de desastres naturais. O Núcleo desenvolveu um banco de dados para armazenar informações referentes a desastres naturais de diferentes formatos e fontes, buscando reunir a ocorrência de eventos que acontecem na região sul do Brasil. O banco de dados possui características espaciais permitindo a localização geográfica dos desastres e a integração com dados cadastrais fornecidos pelos pesquisadores do grupo Geodesastres e pelas entidades parceiras (MORGAN, 2009).

O banco é formado por imagens de satélite, mapas, cartas, documentos, fotos e vídeos relacionados aos eventos de desastres naturais e visa o desenvolvimento de projetos na região sul do Brasil para pesquisa e aplicação de geotecnologias em desastres naturais contribuindo para o mapeamento de áreas de risco. O grupo Geodesastres poderá compartilhar dados com os órgãos que atuam na prevenção e mitigação desses fenômenos e disponibilizar o acesso das informações a pesquisadores e população em geral, porém, isso exige controle em relação às ações possíveis sobre os objetos do banco. Na figura 22 é possível notar quatro ações que o sistema permite: Adicionar, Eliminar, Editar e Exibir, nem todos os membros das entidades parceiras e público em geral podem executar determinadas funções e quando podem estarão sujeitos a determinadas restrições.

Figura 22 - Listagem de fotos inseridas no banco dados geodesastres (Fonte MORGAN 2009).

Em virtude das condições de acesso diferenciadas o CABEC foi integrado ao sistema que manipula o banco de dados de geodesastres a fim de executar o controle de acesso. Considerando que o protótipo do sistema geodesastres foi desenvolvido em PHP e possui uma função chamada *CheckSecurity* que é responsável por avaliar uma tentativa de execução de qualquer ação possível, a mesma foi alterada de modo a invocar o *Web Service* CABEC passando os atributos da autorização para que o serviço faça a avaliação e retorne uma das suas respostas: *Permitido, Negado, Não aplicável ou Indeterminado.* Desta forma o protótipo do sistema geodesastres passou a utilizar o serviço Cabec para controlar o acesso, porém, foi necessário o cadastro prévio de todos os objetos referenciados pelo sistema e suas propriedades contextuais, assim como todos os usuários com suas propriedades contextuais, lembrando que propriedades contextuais foram definidas no capitulo III.

Uma vez cadastradas as propriedades contextuais auxiliam no mapeamento do contexto das entidades que formam o sistema geodesastres tornando-se possível compor expressões de contexto no momento de qualquer solicitação de acesso a recursos. Com esse requisito a próxima etapa foi a construção de políticas de segurança que implementam contextos através de expressões contextuais. Essas políticas determinam se é possível a execução de ações pré-definidas a objetos do banco de dados geodesastres. Algumas dessas políticas são listadas a seguir:

P1: Qualquer Perfil tem permissão de adicionar um evento com suas imagens, desde que faça isso entre as 8:00 e as 22:00 horas.

P2: Para deletar um evento ou uma imagem é necessário o Perfil de Restrito e estar acessando o banco da rede interna do INPE das 8:00 as 18:00 horas.
P3: Somente o Perfil Administrador tem autorização para visualizar imagens de resolução 1200X1000.
P4: Para alterar um evento ou uma imagem é necessário ser proprietário desses objetos ou ter o perfil de Administrador.

Essas políticas foram construídas em XACML e armazenadas no repositório de políticas do serviço CABEC. Como o sistema geodesastres invoca o serviço CABEC para fazer controle de acesso foi possível a realização de testes de funcionalidade e de desempenho do sistema. Os testes de funcionalidade visam demonstrar se o sistema de fato faz o que se propõem fazer em sua especificação e os testes de desempenho permitem uma análise da viabilidade de utilização do serviço em relação ao tempo de resposta do mesmo. A descrição e discussão estão na próxima sessão.

```
<Request>

 <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">

  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>jbandeira</AttributeValue></Attribute>

  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Perfil"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>Parceiros</AttributeValue></Attribute>

 </Subject>

 <Resource>

  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>public.evento</AttributeValue></Attribute
>

 </Resource>

 <Action>

  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>Add</AttributeValue></Attribute>

 </Action>

</Request>
```

Figura 23 – Requisição para adicionar um evento.

```
<?xml version="1.0" encoding="UTF-8" ?>

<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="urn:oasis:names:tc:xacml:1.0:context" xmlns:memos="urn:example:documents" PolicyId="SelectorPolicy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">

<Target> <Subjects>

 <AnySubject/>

 </Subjects>
<Resources><Resource>

<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">

 <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">public.evento</AttributeValue>

 <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" />

 </ResourceMatch>

 </Resource>
 </Resources>

<Actions><Action>

<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">

 <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Add</AttributeValue>

 <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" />

 </ActionMatch>

 </Action> </Actions> </Target>
<Rule RuleId="Descritos" Effect="Permit">

<Condition FunctionId="http://research.sun.com/projects/xacml/names/function#time-in-range">

<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">

 <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#time"
AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time" />

 </Apply>

 <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">08:00:00</AttributeValue>

 <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">22:00:00</AttributeValue>

 </Condition>

 </Rule>

 </Policy>
```

Figura 24 – Política XACML do repositório CABEC

5.3. Testes Funcionais

O primeiro teste funcional levou em conta as credenciais do usuário de código 2 querendo inserir um evento. A requisição da figura 23 foi montada pelo serviço CABEC. Tal requisição se refere ao usuário *jbandeira* de perfil *Parceiros* que pretende *adicionar* um *evento*. No repositório de políticas XACML a política da figura 24 foi encontrada que atendeu a requisição, pois permite que qualquer *Perfil adicione evento* entre as *8:00 e as 22:00* horas. Processando a requisição da figura 23 com base na avaliação da política da figura 24 o serviço CABEC concedeu a resposta verificada na figura 25.

```
<Response>
 <Result ResourceID="public.evento">
  <Decision>Permit</Decision>
  <Status>
   <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
  </Status>
</Result>
</Response>
```

Figura 25. Resposta do serviço Cabec

A resposta é interpretada pelo sistema geodesastres credenciando o usuário a executar a ação pretendida. É importante ressaltar que a política pode ser aperfeiçoada, uma vez que é possível a agregação de mais regras dentro da lógica do CABEC onde várias *expressões de contexto* podem ser combinadas para formular uma política demonstrando a flexibilidade do modelo, portanto, se agregássemos a regra da figura 26 que aponta que o local de acesso dever ser a Rede_interna a resposta muda, pois na montagem da requisição não foi verificado de qual rede (interna, externa) está sendo solicitada. Para identificar a rede o serviço CABEC utiliza as propriedades contextuais que identificam o intervalo que pode corresponder ao rótulo Rede_interna, conforme já foi explicado no capitulo IV.

```
<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Rede_interna</AttributeValue>
      <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                  AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Local"/>
```

Figura 26 - Regra Local: Rede_Interna.

Mais um teste funcional foi realizado com usuário de código 03 tentando acessar o objeto *"public.arquivo_imagem"* com a para leitura. Através dessas credenciais o serviço CABEC gerou a requisição da figura 27. Nessa requisição um usuário com o Perfil *restrito* está tentando acessar um objeto *public.arquivo_imagem* para visualização de dentro da rede interna do INPE.

```
<Request>
 <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Local"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>Rede_interna</AttributeValue></Attribute>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>rceretta</AttributeValue></Attribute>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Perfil"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>Restrito</AttributeValue></Attribute>
 </Subject>
 <Resource>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>public.arquivo_imagem</AttributeValue>
</Attribute>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Zoom"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>1200x1000</AttributeValue></Attribute>
 </Resource>
 <Action>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>Search</AttributeValue></Attribute>
 </Action>
</Request>
```

Figura 27 - Requisição gerada pelo serviço Cabec

A requisição da figura 27 demonstra que o serviço CABEC identificou que o usuário *rcerreta* com perfil *restrito* solicitando a *busca* de um *objeto imagem* de dentro da *rede interna* e que esse objeto tem *resolução de 1200x1000*. No repositório de políticas foi encontrada uma política da Figura 28, essa política permite que o Perfil Restrito acesse objetos do tipo arquivo_imagem com resolução 1200x1000.

```
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy"    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="urn:oasis:names:tc:xacml:1.0:context"     xmlns:memos="urn:example:documents"
PolicyId="SelectorPolicy"     RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-
overrides">
 <Target>   <Subjects>    <Subject>
     <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
 <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Restrito</AttributeValue>
<SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"

                        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Perfil"/>
     </SubjectMatch> </Subject></Subjects><Resources><Resource>
     <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue>
DataType="http://www.w3.org/2001/XMLSchema#string">public.arquivo_imagem</AttributeValue>
       <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"

                        AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>

     </ResourceMatch>
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">

       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">1200x1000</AttributeValue>
       <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"

                        AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Zoom"/>
     </ResourceMatch>
     </Resource>   </Resources>   <Actions>    <Action>

     <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Search</AttributeValue>
      <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"

                       AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>

     </ActionMatch>  </Action>   </Actions>  </Target>

<Rule RuleId="Access" Effect="Permit">
   </Rule>
</Policy>
```

Figura 28 – Política XACML para imagens de alta resolução

A resposta do serviço CABEC será interpretada pelo sistema geodesastres liberando a imagem para visualização do arquivo_imagem conforme pode ser observado na figura 29. É importante notar que na requisição foi identificado que o *usuário rceretta* se encontrava na *rede_interna* no momento em que solicitava a visualização do *arquivo_imagem*, porém, essa informação não é determinante para a avaliação, pois a política não possui restrição a nível de rede, embora pudesse fazê-lo em virtude da a lógica do modelo CABEC permite a agregação dessa propriedade.

```
<Response>
 <Result ResourceID="public.arquivo_imagem">
  <Decision>Permit</Decision>
  <Status>
   <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
  </Status>
 </Result>
</Response>
```

Figura 29. Resposta do serviço CABEC

5.4. Testes de Desempenho

O teste de desempenho foi realizado com lotes de requisições simultâneas, disparadas por um número variado de *threads* com o objetivo de verificar a capacidade de resposta do Cabec dentro de um tempo viável. Para cada grupo de *threads* foi calculado um tempo médio para o atendimento de todas as requisições. O gráfico da figura 30 demonstra o tempo médio para o número de requisições simultâneas variando de 0 a 50.

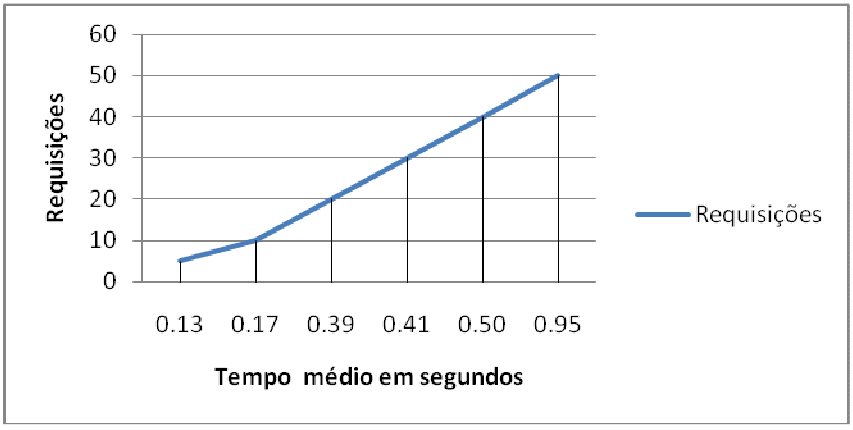

Figura 30. Gráfico de desempenho do Cabec

Para realização dos testes de desempenho foram utilizados dois computadores, um servidor Intel Core 2 Quad 2.33 GHz e com 2 GB de RAM rodando o serviço Cabec e banco de dados Mysql 5.0.77 e um computador cliente Intel Celeron Mobile 1.47 GHz com 1.5 GB de memória RAM. Como pode ser observado no gráfico o tempo médio para atender até 50 requisições ficou abaixo de 1 segundo. Com a adição de mais requisições o tempo médio de resposta tende a aumentar, mas caso exista a demanda de mais de 50 requisições por segundo é possível adquirir um servidor com maior desempenho ou optar por processamento distribuído para garantir um tempo de resposta menor.

5.5. Conclusões do Capítulo

A discussão sobre os modelos semelhantes ao CABEC resultou em uma tabela onde é possível perceber que o modelo CABEC é um artefato diferenciado em relação às características mencionadas. Ele demonstra flexibilidade na construção das políticas de segurança, não depende de outros dispositivos como sensores para seu funcionamento e permite a troca de mensagens via *web services.*

A integração do CABEC com o banco de dados de geodesastres do INPE em Santa Maria revelou sua capacidade de controlar o acesso aos objetos do mesmo através do contexto. Além disso, percebeu-se que embora o banco de dados de geodesastres seja manipulado por um sistema feito com linguagem de programação PHP, e o serviço CABEC

tenha sido implementado em Java, ambos os sistemas foram integrados em função da tecnologia *Web Service.*

O teste de desempenho revelou uma capacidade de resposta do Cabec em média de até um segundo para uma quantia de 50 requisições simultâneas. Caso exista uma demanda maior é possível a instalação do serviço em um servidor mais potente ou optar por processamento distribuído.

6. CONCLUSÃO

Este trabalho apresentou o Modelo de Controle de Acesso Baseado em Expressões de Contexto CABEC, uma extensão RBAC que visa à introdução definições capazes de considerar aspectos dinâmicos do ambiente e combiná-los na construção de regras para formação de políticas de segurança mais abrangentes. Como o RBAC tem em sua credencial de autorização o *usuário*, *objeto* e *ação* o CABEC adicionou a essa credencial a sua definição de *contexto*.

Os aspectos dinâmicos são informações do ambiente (hora, local, data, função, meio físico de acesso...) colhidas no momento da interação com o serviço CABEC na forma de *Propriedades*. As *Propriedades* são dinâmicas, pois podem variar em função do tempo, mas todas são passiveis de representação através das *Expressões de Contexto* que as contém associadas às *Entidades* do ambiente. *Entidades* do ambiente são os elementos ativos na interação, quem está a executando ou quem sofrendo a mesma. A combinação de regras no CABEC é possível pela agregação de mais *Entidades* de contexto dentro de uma *Expressão de Contexto* para torná-la mais abrangente, ou, agregação de mais *Expressões de Contexto* dentro da política de segurança permitindo mais de uma forma de acesso ao mesmo recurso.

A construção de um modelo menos dependente das tecnologias de construção de aplicações foi conseguida através da implementação do CABEC com tecnologias de baixo acoplamento, mais especificamente *Web Services* e *XACML*. Isso permite a troca de mensagens entre aplicações construídas com tecnologias diferentes da qual o CABEC foi construído, uma aplicação construída com a tecnologia PHP pode utilizar o serviço CABEC mesmo ele sendo implementando na tecnologia Java, pois a troca de mensagens ocorre via *Web Service* com formatação *XACML,* padrões aceitos por ambas. Conseqüentemente esse aspecto aumenta o potencial de utilização do modelo.

O *Web Service* CABEC é um artefato de software diferenciado que proporciona melhorar a qualidade da segurança recursos computacionais, sejam eles dados, softwares ou dispositivos, faz isso através do controle de acesso baseado em contexto. Como um modelo de controle de acesso combate a falta de integridade e confidencialidade da informação, isso contribui para aumentar a produtividade uma vez que tem impacto direto na disponibilidade de tais recursos.

O problema das limitações de representação dos modelos que estendem RBAC no que diz respeito a domínios específicos foi resolvido parcialmente com as demonstrações de construção de regras para os domínios de saúde, gerenciamento de redes, ambientes residenciais e de geodesastres. Isso mostra uma abrangência considerável do modelo CABEC tornando-o uma boa opção para a necessidade de controle de acesso em diferentes ambientes.

Um problema encontrado para representação do modelo em forma de políticas XACML foi à falta de uma ferramenta através da qual fosse possível uma interface mais amigável na definição dessas políticas e principalmente que fosse adaptada a forma de representação do contexto definida no modelo CABEC. A falta de consenso entre os modelos que utilizam o contexto na sua forma de representação também foi um problema e ao mesmo tempo um desafio para definir os elementos de modelagem que permitiram a construção do CABEC. Uma padronização nesse sentido ajudaria a evitar confusão nas semânticas de representação desse conceito.

Trabalhos Futuros

Para dar continuidade aos trabalhos originados nessa dissertação, propõem-se os seguintes trabalhos futuros:

- Adaptar o Sistema de Gerenciador de Políticas de Controle de Acesso SGPCA para construção de políticas de acordo com os elementos do modelo CABEC. Esse sistema já permite a construção de políticas em XACML com as quais é possível representar contextos temporais e de localização, uma adaptação ao CABEC aumentaria o nível de abrangência de construção dessas políticas e proporcionaria uma interface mais amigável ao gerente de segurança. Além disso, o SGPCA faz o controle de conflitos entre as políticas, evitando que para um mesmo sistema exista uma política positiva e outra negativa para a mesma situação.
- Aperfeiçoar o modelo CABEC para suportar separação de responsabilidades e hierarquia de perfis e delegação de responsabilidades. Os dois primeiros conceitos são suportados pelo modelo RBAC, porém não são tratados pela versão atual do CABEC que estende o RBAC somente em seu nível básico, já a delegação de responsabilidades foi implementada no modelo CIBAC e pode ser adaptada para o modelo CABEC.
- No âmbito da arquitetura *Web Service* existem algumas vulnerabilidades como: Manipulação de parâmetros, espionagem de rede e repetição de mensagens. *WS-Security*,

possibilita o uso de artefatos de segurança na abordagem desses problemas: Assinaturas digitais e criptografia de documentos XML. Sugere-se como trabalho futuro, testar a eficiência desses artefatos diante das vulnerabilidades apresentadas implementando os mesmos no serviço CABEC.

- Testar o serviço CABEC no âmbito do controle de acesso para redes wireless durante as aulas na UFSM. O teste do serviço CABEC nesse ambiente pode ser relevante para melhor entendimento do seu comportamento e demonstração de sua abrangência.

REFERÊNCIAS

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

BACON, J.; MOODY, K.; YAO, W. A Model of OASIS Role-Based Access Control and its Support for Active Security. *ACM Transaction on Information and System Security*, v.5, p.492-540, nov. 2002.

BERTINO, E.; BONATTI, P. A.; FERRARI, E. TRBAC: A Temporal Role-Based Access Control Model. *ACM Transaction on Information and System Security*, 4(3):191–233. 2001.

BERTINO E., GERTZ M. Security and policy for Geospatial Data: Concepts and Research Directions. *Inaugural Paper for the ACM SPRINGL Workshop,* 2008, CA USA.

BRINKLEY, D. L.; SCHELL,R.R. Concepts and terminology for computer security. In: ADRAMS, M. D.; JAJODIA, S.; PODELL, H. J. (Ed). *Information security:* an integrated collection of essays. Los Alamitos, CA IEEE Computer Society Press, jan 1995. P. 40-97.

COVINGTON, M. J.; SASTRY, M. R. A Contextual Attribute-Based Access Control Model. *Corporate Technology Group. Intel Corporation*. January, 2006.

COVINGTON, M. J.; FOGLA, P.; ZHAN, Z.; AHAMAD, M. A Context-Aware Security for Emerging Applications. *In : 18TH Annual Computer Security Applications Conference,*2003.

COVINGTON, M. J.; LONG, W.; SRINIVASAN, S.; DEY, A. K.; AHAMAD, M.; ABOWD, G. D. Securing context-aware applications using environment roles. *In: Sixth ACM Symposium on access control models and technogogies, 2001*. Proceedings p. 10-20.

DAMIANI, L, M.; BERTINO, E.; GEO-RBAC: a Spatially Aware RBAC. *ACM Transactions on Information Systems and Security,* Vol. 00, No. 00, 2006. Páginas 1 a 34.

DEY, K. AND ABOWD, D. (1999) Towards a Better Understanding of Context and Contextawareness. *In: Gvu technical report GIT-GVU-99-22,* College of Computing, Georgia Institute of Technology.

FERRAIOLO, D. F.; SANDHU, R.; GAVRILA, S.; KUHN, D. R.; CHANDRAMOULI, R. Proposet NIST standard for role-based access control. *ACM Transactions on Information and System Security*, v. 4, n.3, p. 224-274, ago. 2001.

HANAUER G. WS Security – Estudo e Verificação da Qualidade de Proteção em Web Services. Monografia Ciências da Computação. Florianópolis 2004.

HIGASHIYAMA M. JACOWEB-ABC: Integração do Modelo de Controle de Acesso UCONABC no CORBASEC. Dissertação de Mestrado do Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná. Curitiba, 2005.

GEORGIADIS, C. K.; MAVRIDIS, I.; PANGALOS, G.; THOMAS, R. Flexible team-based access control using contexts. *In: Sixth ACM Symposium on Access Control Models and Technologies.* Proceedings p. 21-27, 2001.

JOSHI, J. B. D.; AREF, W. G.; GHAFOOR, A.; SPAFFORD, E. H. Security models for web-based applications. *Communications of the ACM*, v. 44, n. 2, p. 38-44, fev. 2001.

JOSHI, J. B. D.; BERTINO, E.; SHAFIQ, B.; GHAFOOR, A. Dependencies and separation of duty constraints in GTRBAC**.** *In: Eighth ACM Symposium on Access Control Modelsand Technologies*, Proceedings p.51-64, 2003.

LIN, A.; BROWN, R. The application of security policy to role-based access control and the common data security architecture. Computer Communications, v.23, p.1584-1593,2000.

LIMA, Paulo Ricardo Barbieri Dutra, NUNES, Raul Ceretta, SOARES, Gerson Antunes. Sistema Gerenciador de Políticas de Controle de Acesso. *XXVII Encontro Nacional de Engenharia de Produção. Foz do Iguaçu*, 2007.

LONGSTAFF, J.; LOCKYER, M.; NICHOLAS, J. The Tees confidentiality model: an authorisation model for identities and roles. In: Eighth ACM Symposium on Access Control Models and Technologies, *Proceedings* p. 125-133, 2003.

KUMAR, A.; KARNIK, N.; CHAFLE, G. Context sensitivity in role-based access control. *ACM SIGOPS Operating Systems Review*, v.36, n.3, p.53-66, jul. 2002.

MARTINS, R M. Composição Dinâmica de Web Services. Dissertação apresentada à Universidade do Vale do Rio dos Sinos no programa de Computação Aplicada. São Leopoldo, 2007.

MCDANIEL, P.; On Context in Authorization Policy**.** SACMAT'03, Junho de 2003, Como, Itália.

MOTTA, G. H. M. B.Um modelo de autorização contextual para o controle de acesso ao Prontuário Eletrônico do Paciente em Ambientes Abertos e Distribuídos. *Tese de doutorado apresentada à Escola Politécnica da Universidade de São Paulo*, São Paulo-SP, Brasil. 2003.

MORGAN, J.;SAUSEN, T. Banco de dados para o Núcleo de Pesquisa e Aplicação de Geotecnologias em Desastres Naturais e Eventos Extremos do Centro Regional Sul de Pesquisas Espaciais do INPE. *Escola Regional de Banco de Dados da Região Sul*, Abril 2009.

NEUMANN, G.; STREMBECK, M. An approach to engineer and enforce context constraints in an RBAC environment. *In: Eighth ACM Symposium on Access Control Models and Technologies,* Proceedings p. 65-79, 2003.

NETO, J. Divulgação do Orçamento Público via Web Services em Java. *Dissertação apresentada no programa de Mestrado em Engenharia Elétrica*. Recife, 2004.

OASIS XAML, eXtensible Access Control Markup Language (XACML) Version 1.0. *Manual da linguagem de marcação XACML, disponível em* http://www.oasis-open.org/committees/xacml/repository/, acessado em Out. 2009.

OASIS. Web Services Security: Soap Message Security 1.0. Disponível em: http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soapmessage-security-1.0.pdf, acessado em Nov. 2009.

OSBORN, S.; SANDHU, R.; MAUNAWER, Q. Configuring role-based access control to enforce mandatory and discriminatory access control policies. *ACM Transactions on Information Systems Security*, v. 3, n. 2, p. 85-106, mai. 2000.

PASCOE, J. Adding generic contextual capabilities to wearable computers. *In:International Symposium on Wearable Computers*, pp. 92–99, 1998.

SAMUEL, A.; GHAFOOR, A.; BERTINO E. Context-Aware Adaptation of Access-Control Policies *IEEE Computer Society*, pages:51-54 February, 2008.

SOMMERVILLE, I. Software Engineering**.** 5. ed. Addison-Wesley. 1996.

SANDHU, R. S; SAMARATI, P. Access Control: principles and pratice. *IEEE Communications Magazine*, p. 40-48, set. 1994.

SANDHU, R. S.; COYNE, E. J.; YOUMAN, C. E. *Roled-based access control models. IEE Computer*, P. 38-47, fev. 1996.

SOARES, G. A.; NUNES, R. C. & AMARAL, E. M. H. do. *Um Modelo de Controle de Acesso Baseado em Contexto para Autorizações a Informações Médicas*. In: XXXII Conferência Latino-Americana de Informática, Santiago do Chile, 2006.

SCHILIT, B.; THEIMER, M. Disseminating active map information to mobile hosts. *n:IEEE Network*, v.8, n.5, pp. 22–32, 1994.

STALLINGS, W. *Cryptography and network security:* principles and practice. 2. Ed. Prentice Hall, 1999. 569 p.

WILIKENS, M.; FERITI, S.; SANNA, A.; MASERA, M. A context-related authorization and access control method based on RBAC: a case study from the health care domain**.** *In: Seventh ACM Symposium on Access Control Models and Technologies*, Proceedings p.117-124. 2002.

WATHIER, J, A.; Segurança em Web Services com WS-Security. *Monografia apresentada ao curso de Especialização em Desenvolvimento, Segurança e qualidade na Internet na Universidade Federal do Rio Grande do Sul, como requisito parcial para obtenção do titulo de especialista*. Porto Alegre 2005.

VIJAYALAKSHMI A.; SOON A. C. An Authorization Model for Geospatial Data. *IEEE Transactions on dependable and secure computing, vol. 1, no. 4, october-december 2004*

KULKARNI D.; TRIPATHI A. *Context-Aware Role-based Access Control in Pervasive Computing Systems SACMAT'08,* June 11–13, 2008, Estes Park, Colorado, USA. Copyright 2008 ACM 978-1-60558-129-3/08/06

W3 CONSORTIUM. Extensible Markup Language (XML). *2. ed., out. 2000 Disponível em <http://www.w3.org/xml/>. Acesso em: dezembro 2009.*

WEB SERVICES, web services architecture requirements, 2004. *Disponível em < http://www.w3.org/TR/wsa-reqs/>. Acesso em: dezembro 2009.*