Um Modelo de Autorização Contextual para o Controle de Acesso Baseado em Papéis

Gustavo H. M. B. Motta[1, 2, 3] e Sérgio S. Furuie[1, 2]

[1] Instituto do Coração – Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo,
[2] Escola Politécnica da Universidade de São Paulo, [3] Departamento de Informática – Universidade Federal da Paraíba
e-mail: gustavo.motta@incor.usp.br

Resumo: *Este trabalho propõe um modelo de autorização contextual que estende e refina o modelo de referência para controle de acesso baseado em papéis do NIST (National Institute of Standards and Technology). O modelo suporta autorizações positivas e negativas, sobreposição de autorizações herdadas para definição de exceções, ativação automática de papéis, separação de responsabilidades estática e dinâmica baseadas em conflitos fortes e fracos entre papéis. Uma autorização contextual será positiva ou negativa de acordo com a avaliação uma regra de autorização. Tal regra é especificada como uma expressão lógica que relaciona informações ambientais (contextos) sobre o usuário corrente, momento e local de acesso, dentre outros dados que podem ser livremente programados e disponibilizados para definição de políticas de acesso mais complexas.*

Palavras-chave: *autorização contextual, controle de acesso baseado em papéis, segurança*

Abstract: *This work proposes a contextual authorization model that extends and reifies NIST (National Institute of Standards and Technology) role-based access control reference model. The model supports positive and negative authorizations; authorization overriding to set exceptions; automatic role activation; static and dynamic separation of duty based on weak and strong role conflicts. Contextual authorizations are positive or negative according to evaluation of an authorization rule, which is specified as a logical expression. Such expression relates environmental information (contexts) about current user, date/time and location of access, among other data that can be programmed and incorporated to define a more complex access policy.*

Keywords: *contextual authorization, role-based access control, security*

1. Introdução

Os recentes avanços nas tecnologias de comunicação e computação viabilizam, técnica e economicamente, a disponibilidade em larga escala de uma variedade de informações, independente do local e do momento do acesso. Entretanto, isso acarreta problemas em situações onde restringir o acesso é imperativo para resguardar a privacidade de indivíduos e instituições, como em aplicações na área de saúde ou aquelas envolvendo negócios nos setores público e privado.

Modelos tradicionais para controle de acesso, como os discricionário e compulsório, carecem dos requisitos necessários para definição e administração viável de políticas de acesso, particularmente em aplicações corporativas emergentes, que demandam um controle com granularidade fina para um grande número de usuários e recursos [(3)]. O controle de acesso baseado em papéis (CABP)[(2)] tem características mais adequadas para tais aplicações e vem recebendo considerável atenção como uma alternativa viável. No CABP, autorizações de acesso são associadas a papéis, com um usuário possuindo as autorizações dos papéis a que pertence. Papéis atribuídos a um usuário denotam funções que descrevem a autoridade e a responsabilidade concedidas ao mesmo numa organização. Isto facilita a administração da política de acesso, pois permite colocá-la na perspectiva de um modelo organizacional [(9)]. Ademais, o CABP é politicamente neutro, podendo suportar os modelos compulsório ou discricionário, dentre outros [(3 e 10)].

O padrão para CABP proposto pelo NIST [(2)] (*National Institute of Standards and Technology*) estabelece um modelo de referência que deixa em aberto a representação concreta e a interpretação de autorizações, cabendo esta tarefa a modelos mais específicos. Este trabalho propõe um modelo de autorização contextual que estende e refina o modelo de CABP do

NIST. O uso de contextos permite que a política de acesso seja estabelecida com base em fatores dinâmicos existentes no momento em que uma solicitação de acesso é realizada. Esse é um importante requisito[11] para o controle de acesso em aplicações na área de saúde, onde o uso das autorizações contextuais é investigado, mas que não é contemplado no modelo de referência. Por exemplo, num hospital, médicos são autorizados a prescrever medicamentos a pacientes, mas apenas para aqueles internados ou atendidos na sala de emergência. Assim, uma autorização contextual é "ciente" do contexto (pacientes internados, local do acesso) associado a operação a realizar (prescrever medicamento), possibilitando a definição de políticas de acesso mais precisas, flexíveis, com baixo nível de granularidade, onde o acesso é concedido ou negado no momento exato, de acordo com a necessidade do usuário.

O trabalho está organizado da seguinte forma. A seção 2 resume o modelo de referência para CABP do NIST. A seção 3 descreve o modelo de autorização contextual proposto para o CABP. A seção 4 destaca aspectos relevantes de sua implementação e finalmente a seção 5 conclui o trabalho.

2. O Controle de Acesso Baseado em Papéis do NIST

O padrão NIST para CABP [2] (Figura 1) possui quatro conjuntos de entidades principais: *U* (usuários), *P* (papéis), *A* (autorizações) e *S* (sessões). Especifica que uma autorização é um relacionamento *n* para *m* entre os recursos protegidos (objetos) e respectivas formas de acesso (operações), mas deixa em aberto a representação de usuários, papéis, objetos e operações, bem como a interpretação de autorizações, cabendo estas tarefas a modelos mais detalhados. Estas entidades possuem os seguintes relacionamentos: usuário-papel *UP*; papel-autorização *PA*; hierarquia de papéis *HP* e sessões. As relações *UP* e *PA* especificam associações *n* para *m* entre usuários e papéis; e entre papéis e autorizações, respectivamente. *HP* define uma relação de ordem parcial entre papéis, dispondo-os em hierarquias a fim de melhor representar as linhas de autoridade e responsabilidade de uma organização. Uma sessão se relaciona com um único usuário por vez, mas permite que ele assuma (ative) múltiplos papéis simultaneamente, desde que estes papéis estejam associados ao usuário na relação *UP*. Por outro lado, um usuário pode ter várias sessões ao mesmo tempo.

Aos relacionamentos do padrão, podem-se estabelecer restrições para minimizar as chances de fraude ou dano acidental pela demasiada concentração de poder numa única pessoa. Uma restrição típica é limitar o número máximo de papéis de um usuário. Outra é a *separação de responsabilidades* (SR), que distribui a responsabilidade para realização de uma ação por múltiplos usuários, de modo que uma pessoa não seja poderosa o suficiente para efetuá-la sem um conluio. A SR é definida através de papéis mutuamente exclusivos, tanto na relação *UP*,

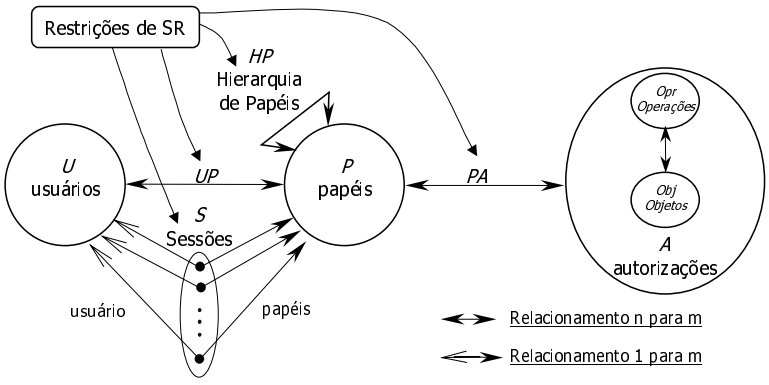

Figura 1 – Padrão NIST de referência para CABP [2]

quanto na relação *PA*. Em *UP*, dois ou mais papéis mutuamente exclusivos não podem ter usuários em comum associados. Já em *PA*, define-se a separação de responsabilidades proibindo-se a associação de uma mesma autorização a papéis mutuamente exclusivos. A idéia é adotá-la para reduzir a possibilidade de um usuário assumir papéis onde ocorram conflitos de interesse. Quando a restrição é imposta no momento em que estas relações são estabelecidas, ela é denominada de *separação de responsabilidades estática* (SRE). A *separação de responsabilidades dinâmica* (SRD) ocorre quando potenciais conflitos de interesse são detectados no momento em que um usuário tenta ativar mais de um papel simultaneamente, independente das sessões que abriu. A SRD admite um usuário possuir vários papéis conflitantes, desde que não sejam ativados ao mesmo tempo.

3. Modelo de Autorização Contextual

O modelo de autorização contextual proposto nesta seção refina e estende o padrão NIST para CABP visto anteriormente. É uma evolução do trabalho [4 e 5] desenvolvido no InCor (Instituto do Coração do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo.) para atender os requisitos de controle de acesso para o prontuário eletrônico do paciente. O modelo suporta a hierarquização de papéis restrita à estrutura de árvores invertidas, com herança das autorizações associadas aos papéis mais gerais pelos papéis descendentes destes, mais específicos. Inclui ainda autorizações positivas e negativas; sobreposição de autorizações herdadas para estabelecimento de exceções; separação estática e dinâmica de responsabilidades baseada em conflitos fortes e fracos entre papéis; ativação automática de papéis.

Uma autorização contextual será positiva, concedendo o acesso, ou negativa, proibindo o acesso, com base na avaliação de uma expressão lógica, denominada de *regra de autorização*, no momento da solicitação de acesso. Essa expressão é definida em termos de variáveis ambientais (contextos) que, quando avaliadas, resultam em informações sobre o usuário corrente, data/hora do acesso, informações de rede (local do acesso) e outras que podem ser livremente programadas e incorporadas para especificação de políticas de acesso mais complexas.

A fim de facilitar o entendimento, a subseção 3.1 descreve o modelo de autorização sem contextos, que são introduzidos na subseção 3.2 em conjunto com as regras de autorização.

3.1. Autorizações

Uma autorização de acesso é uma tupla $<p, obj, tp, opr, ta>$, onde p é o papel para o qual um privilégio é estabelecido; obj especifica o recurso para o qual o privilégio se aplica; tp especifica o tipo de privilégio, positivo (+) quando concedido e negativo (−) quando proibido; opr é o privilégio de acesso estabelecido e ta especifica se o tipo de autorização é $forte$ ou $fraca$.

Sobreposição de Autorização – Com os papéis organizados numa estrutura de árvore invertida, autorizações associadas a papéis mais genéricos são herdadas em papéis descendentes destes. Porém, estas autorizações herdadas podem ser sobrepostas de acordo com a seguinte definição:

- Uma autorização $<p_1, obj_1, tp_1, opr_1, ta_1>$ sobrepõe uma outra autorização $<p_2, obj_2, tp_2, opr_2, ta_2>$ se, e somente se, p_1 é descendente de p_2 e $obj_1 = obj_2$ e $opr_1 = opr_2$.

Uma sobreposição estabelece uma exceção quando o tipo do privilégio (+ ou −) da autorização herdada é modificado na autorização que sobrepõe, isto é, quando $tp_1 \neq tp_2$. A fim de disciplinar o uso de exceções, para cada autorização concedida, + ou −, especifica-se se ela admite exceções ou não. As autorizações do tipo $fraca$ admitem exceções, ao passo que, as do tipo $forte$, não as admitem. Uma autorização forte estabelece uma política de acesso absoluta, que

não tolera contradições, prevalecendo portanto sobre qualquer autorização equivalente do tipo fraca. Autorizações fortes e fracas foram originalmente introduzidas por Bertino et al.[1] num mecanismo de autorização para sistemas de gerência de bancos de dados relacionais.

Separação de Responsabilidades – É especificada neste modelo com base nos conflitos existentes entre as autorizações, mas de modo natural e não arbitrário. Isto porque as autorizações positivas e negativas sinalizam conflitos de interesse no acesso a um determinado recurso. Se para um papel, um acesso é autorizado para um recurso e, em outro papel, o mesmo acesso é contradito, então certamente haverá conflitos para um usuário exercendo ambos os papéis. Assim, os conflitos são deduzidos automaticamente segundo a autoridade e a responsabilidade estabelecidas para cada papel através das autorizações associadas.

O conflito entre autorizações ocorre estaticamente quando se estabelecem as associações entre autorizações e papéis e dinamicamente quando um usuário ativa mais de um papel simultaneamente. As definições de cada tipo de conflito seguem abaixo:

- *Conflito estático*: duas autorizações $A_1 = <p_1, obj_1, tp_1, opr_1, ta_1>$ e $A_2 = <p_2, obj_2, tp_2, opr_2, ta_2>$ conflitam estaticamente se, e somente se, A_1 estabelece uma exceção para A_2 e $ta_1 = ta_2$;
- *Conflito dinâmico*: duas autorizações $<p_1, obj_1, tp_1, opr_1, ta_1>$ e $<p_2, obj_2, tp_2, opr_2, ta_2>$ conflitam dinamicamente se, e somente se, para $p_1 \neq p_2$, p_1 é ativado simultaneamente com p_2 para um mesmo usuário e $obj_1 = obj_2$ e $tp_1 \neq tp_2$ e $opr_1 = opr_2$ e $ta_1 = ta_2$.

Quando o tipo de autorização é forte em autorizações conflitantes (dinâmica ou estática), o tipo de conflito é denominado de *conflito forte*. Caso contrário, o conflito é denominado *conflito fraco*. Dois ou mais papéis que possuam autorizações conflitantes entre si são denominados *papéis conflitantes*. Nota-se que, quando os tipos da autorização são diferentes, não há conflito, pois autorizações fortes prevalecem sobre as autorizações fracas.

Nesse modelo, a separação de responsabilidades estática não admite autorizações que estabeleçam conflitos estáticos fortes entre papéis, pois, numa mesma linha de responsabilidades na hierarquia, não pode haver contradição nas ações permitidas (ou proibidas) de modo absoluto através de uma autorização forte. Por outro lado, o conflito estático fraco é admitido em papéis distintos e a política de resolução dos conflitos dá-se da seguinte maneira:

- Uma autorização fraca, negativa ou positiva, especificada num papel, prevalece sobre qualquer autorização fraca conflitante especificada em papéis ascendentes deste;

Já a separação de responsabilidades dinâmica proíbe a ocorrência de conflitos dinâmicos fortes. Um usuário pode possuir papéis que eventualmente estabeleçam conflitos dinâmicos fortes, porém é proibida a ativação simultânea destes papéis para ele, evitando-se a ocorrência dos conflitos. Suponha que um usuário tenha associado os papéis *Médico* e *Pesquisador*, que possuem conflitos dinâmicos fortes entre si. O usuário poderá ativar cada um deles isoladamente, mas não simultaneamente. Já os conflitos dinâmicos fracos são admitidos e política de resolução dá-se da seguinte maneira:

- Havendo duas ou mais autorizações fracas conflitantes dinamicamente, prevalecerá aquela autorização que concede o acesso.

Ativação de Papéis – Considerando que usuário abriu uma ou mais sessões após uma autenticação bem sucedida, um papel inicial deverá ser ativado para ele. Dependendo do caso, o usuário escolhe explicitamente o papel inicial, ou então um papel *default* é ativado. Daí em diante, papéis subseqüentes poderão ser ativados automaticamente de acordo com a necessidade do usuário utilizar um recurso, tomando-se por base a idéia proposta por Obelheiro et al.

(6) para ativação automática de papéis. Entretanto, esta ativação deve satisfazer as condições definidas a seguir:

Condições para Ativação de Papéis – Um usuário possui um conjunto de papéis ativos (P_A), um conjunto de papéis disponíveis (P_D) e o conjunto de papéis associados estaticamente (P_{AE}), obtido da relação UP (Figura 1). Antes de iniciar a primeira sessão, o conjunto P_A é vazio e os conjuntos P_D e P_{AE} são iguais e correspondem aos papéis associados ao usuário. As seguintes assertivas devem prevalecer a qualquer momento (invariante do estado do sistema para um usuário):

- $P_A \cup P_D \subseteq P_{AE}$;
- $P_A \cap P_D = \varnothing$;
- $\forall\ p_a \in P_A; p_d \in P_D \Rightarrow p_a$ não tem autorizações que conflitam fortemente com as de p_d.

Após o início da primeira sessão, o conjunto P_A é inicializado com o papel inicial escolhido para o usuário. A partir desse momento, o conjunto de papéis disponíveis para ativação (P_D) corresponderá aos papéis presentes em P_{AE}, com exceção daqueles que conflitem fortemente com algum papel ativo presente em P_A. Assim, não há possibilidade da ativação simultânea de dois papéis que conflitam fortemente para um mesmo usuário. A desativação dos papéis só ocorre quando todas as sessões que o usuário abriu são encerradas. Estas condições vigoram independente das sessões do usuário.

Por exemplo, suponha um usuário com P_{AE} = {*Médico*, *Pesquisador*, *Diretor*}, onde os papeis *Médico* e *Pesquisador* conflitam fortemente. Ativando-se inicialmente o papel de *Médico*, com P_A = {*Médico*}, não se poderá ativar simultaneamente o papel *Pesquisador*, pois como estes dois papéis conflitam, o conjunto de papéis disponíveis, será P_D = {*Diretor*}, segundo as condições estabelecidas anteriormente. Para ativar o papel *Pesquisador*, deve-se desativar o papel *Médico*, de modo que P_A volte a ser vazio e P_D = {*Médico*, *Pesquisador*, *Diretor*}. Assim, o usuário poderá assumir um novo papel inicial, no caso, *Pesquisador*. Em qualquer caso, o papel *Diretor* poderia ser ativado por não ter conflito forte com os outros papéis.

Se existir em P_D um papel que autorize um acesso a um recurso pretendido pelo usuário, o mesmo será ativado automaticamente caso não existam papéis ativos em P_A que autorizem o acesso ao recurso ou neguem o acesso através de uma autorização forte. O modelo, porém, não exclui a possibilidade do usuário ativar explicitamente um papel. A conveniência de adotar tal estratégia vai depender da política de controle de acesso desejada.

3.2. Integrando Contextos e Regras às Autorizações

Contextos – Um contexto denota informações ambientais existentes no momento de uma solicitação de acesso. Variáveis contextuais típicas trazem informações sobre o usuário corrente (nome de *login*, matrícula, unidade de lotação, papéis associados, etc.), momento do acesso (data, hora, dia da semana, etc.), local da solicitação de acesso (endereços IP, DNS e porta do cliente) e informações de segurança (domínios DNS confiáveis, indicação se a conexão é segura, etc.). Contextos ainda podem indicar informações mais específicas, relacionadas ao recurso que se pretende acessar via aplicação utilizada pelo usuário. Por exemplo, um contexto poderá representar o conjunto dos pacientes internados num hospital, ou indicar se um paciente está sendo acompanhado por um determinado médico ou equipe médica.

Neste modelo, informações contextuais são obtidas através de uma interface de programação, cuja definição em CORBA IDL (*Interface Definition Language*) é apresentada a seguir:

```
module Contexts {
  interface Context {
    Any getValue (in string index);
    boolean inEvaluation(in Any element, in string aSetName);
    Any functionApplication(in string functionName, Vector argumentList);
  };
};
```

Cada contexto (usuário, momento de acesso, local de acesso, segurança e outros livremente programados) deve implementar a interface `Context` acima. A operação `getValue` retornará um objeto do tipo `Any` (permite valores de qualquer tipo), associado ao *string* do parâmetro de entrada `index`, cujo valor denota o nome de uma variável contextual. Por exemplo, as variáveis contextuais associadas ao usuário corrente, como "login" e "matrícula" podem ser implementadas nesta operação, retornando valores dos tipos *string* e inteiro, respectivamente. A segunda operação – `inEvaluation` – é usada para implementar a operação que testa a pertinência de um elemento – `element` – num conjunto denotado pelo valor de `aSetName`. Por exemplo, a verificação se um paciente está internado pode ser implementada nesta operação. Assim, `element` indicaria a chave primária do paciente e `aSetName` determinaria a submissão de uma consulta em SQL (*Structured Query Language*) para verificar se o paciente está internado. Em caso afirmativo, o valor booleano verdadeiro seria retornado, caso contrário, o valor falso. Finalmente, a operação `functionApplication` permitirá a implementação de contextos mais complexos através de chamadas de funções, denotadas pelo valor do parâmetro `functionName`, com lista de argumentos denotada pelo parâmetro do tipo vetor (seqüência de valores do tipo `Any`) chamado `argumentList`.

Regras – Relacionam as informações ambientais dos contextos em expressões lógicas que especificam uma política de acesso para um recurso protegido. São definidas numa linguagem de expressões lógicas capaz de acessar as operações implementadas nos contextos e relacioná-las através de operadores aritméticos (+, –, *, /, % - módulo), de conjunto (`in` - pertinência), relacionais (>, <, >=, <=, =, !=) e booleanos (&, |, !). Permite ainda a chamada de funções implementadas nos contextos e a definição de regras parametrizadas, de modo que aplicações que solicitam autorizações de acesso possam passar argumentos para regras. A regra abaixo

```
exp-abs(umCodPac) {
  umCodPac in pacCtx.pacientes_internados |
  netCtx.peer_dns in secCtx.dominios_emergencia
}
```

é um exemplo de uma expressão parametrizada. O parâmetro `umCodPac` indica o código de identificação de um paciente. Três contextos são utilizados no corpo da expressão: `pacCtx`, contexto de pacientes; `netCtx`, contexto de rede e `secCtx`, contexto de segurança. Variáveis, conjuntos ou funções implementados num contexto são acessadas usando o `<nome do contexto>.<nome>`. No exemplo, `pacCtx.pacientes_internados` e `secCtx.dominios_emergencia` denotam conjuntos onde o operador de pertinência `in` é implementado pela operação `inEvaluation` em cada contexto. No primeiro caso, verifica se o código passado como parâmetro é de um paciente internado. No segundo caso, verifica se a variável `peer_dns` (domínio DNS de onde a solicitação de acesso é realizada) do contexto `netCtx` pertence ao conjunto dos domínios válidos para a sala de emergência. A expressão é avaliada como verdadeira se o paciente identificado por `umCodPac` está internado ou se o acesso é realizado da sala de emergência.

Autorizações Contextuais – Uma autorização contextual estende o modelo definido pela tupla `<p, obj, tp, opr, ta>`, permitindo que o tipo de privilégio `tp` seja uma regra que, quando

avaliada para verdadeiro, resulte numa autorização positiva (+), e, em caso contrário, resulte numa autorização negativa (–). Logo, a regra do exemplo anterior poderia compor uma autorização contextual da seguinte forma:

```
< Médico, Prontuário,
  exp-abs(umCodPac) { umCodPac in pacCtx.pacientes_internados |
                      netCtx.peer_dns in secCtx.dominios_emergencia },
  PrescreverMedicamento, Fraca >
```

Esta autorização especifica a seguinte política de acesso: *médicos são autorizados a prescrever medicamentos no prontuário do paciente somente se este estiver internado ou sendo atendido na sala de emergência*. A aplicação passa o código do paciente como argumento da autorização no momento em que solicita a autorização de acesso.

Para evitar a ocorrência de conflitos dinâmicos fortes, a utilização de regras só será permitida em autorizações do tipo `fraca`. Assim, as condições para ativação de papéis não serão violadas. Isto porque, na hipótese de haver regras em autorizações do tipo `forte`, não seria possível determinar em geral, *a priori*, se dois papéis possuem autorizações que conflitam fortemente.

4. Aspectos de Implementação

Atualmente, uma implementação do modelo proposto vem sendo usada no InCor para autorização de acesso ao prontuário eletrônico do paciente na *intranet* da instituição e para autorização de acesso a 15 aplicações médico-hospitalares . Um total de 1232 usuários tem hoje acesso controlado a estas aplicações com diferentes privilégios, dependendo dos papéis associados a cada um deles, dentre os 45 papéis definidos. A intenção é adaptar paulatinamente as aplicações existentes e as novas aplicações para esse esquema de autorização de acesso, de modo que todas as aplicações institucionais tenham seu acesso controlado segundo uma política unificada e consistente.

A representação das entidades usuários, papéis, autorizações contextuais e respectivos relacionamentos é armazenada num serviço de diretórios hierarquizado, cujo acesso e esquemas de descrição de dados são padronizados pelo protocolo LDAP[12]. A gerência das sessões dos usuários, do mecanismo de ativação de papéis, do serviço de autorização de acesso e de autenticação de usuários foram implementados na linguagem de programação Java e integrados a um servidor CORBA/Visibroker. A autenticação do usuário e o serviço de autorização são oferecidos às aplicações clientes de forma padronizada através das interfaces apropriadas do CORBA *Security Service*[7] e do CORBA *Resource Access Decision Facility*[8], respectivamente. Interfaces de contextos são implementadas em Java ou acessadas em Java via CORBA IDL e disponibilizadas em pacotes de extensão Java (*plug-in*). Assim, novos contextos podem ser introduzidos automaticamente ao servidor sem necessidade de modificá-lo.

Uma aplicação Java foi desenvolvida para a administração do modelo no servidor LDAP, permitindo o cadastramento de usuários, papéis, recursos protegidos (objetos e operações), bem como a configuração das autorizações contextuais. Esta administração só é permitida para usuários privilegiados que possuem o papel de administrador.

5. Conclusão

Este trabalho propôs um modelo de autorização contextual para CABP que suporta hierarquia de papéis numa estrutura de árvores invertidas com herança de autorizações; autorizações positivas e negativas; sobreposição de autorizações herdadas para estabelecimento de exceções; separação de responsabilidades estática e dinâmica baseadas em conflitos fortes e fracos entre papéis. O modelo permite estabelecer, desde as políticas de acesso mais estritas, até as mais

permissivas, com a variação entre os dois extremos regulada pela especificação de autorizações fortes e fracas. Outra vantagem é a obtenção natural da separação de responsabilidades, estática e dinâmica, a partir das ações permitidas ou proibidas para um usuário. Esta é uma característica importante quando é necessário administrar uma política de acesso em corporações com grande número de usuários, recursos protegidos e papéis. Já a ativação automática de papéis facilita para o usuário final usar os recursos protegidos de acordo com a necessidade e o direito de utilizá-los, sem a preocupação com a ativação explícita de papéis.

Ademais, a integração nas autorizações de regras baseadas em informações contextuais, dependentes do usuário, data/hora, local e outras que podem ser livremente programadas, confere grande flexibilidade e poder de expressividade na especificação de políticas de acesso. Como conseqüência, o controle de acesso pode ser efetuado de forma mais precisa, no momento exato, de acordo com a necessidade e o direito do usuário, mas sem custos gerenciais excessivos. Isto porque lógicas de controle de acesso mais complexas, tradicionalmente embutidas nas aplicações protegidas, podem agora ficar isoladas destas aplicações, embora mantendo o contexto necessário para conceder ou não uma autorização. Assim, mudanças na lógica de autorização não implicam em mudanças nas aplicações.

O uso da implementação do modelo proposto para atender as necessidades de controle de acesso para o prontuário eletrônico do paciente no InCor[4 e 5] visa ao seu aperfeiçoamento e validação. Um trabalho futuro indispensável será o aprimoramento do modelo de administração da política de acesso existente, hoje centralizado no papel *Administrador*, para facilitar o uso prático do modelo proposto em ambientes reais, nesse e em outros domínios de aplicação.

6. Referências

1. Bertino, E.; Jajordia, S. e Samarati, P. “A Flexible Authorization Mechanism for Relational Data Management Systems”, *ACM Transactions on Information Systems* 17, 2 (Abril 1999), 101-140.
2. Ferraiolo, D. F.; Sandhu, R.; Gavrila, S.; Kuhn, D. R. e Chandramouli, R. “Proposed NIST Standard for Role-Based Access Control”, *ACM Transactions on Information and System Security* 4, 3 (Agosto 2001), 224-274.
3. Joshi, J. B. D.; Aref, W. G.; Ghafoor, A. e Spafford, E. H. “Security Models for Web-Based Applications”, *Communications of the ACM* 44, 2 (Fevereiro 2001), 38-44.
4. Motta, G. H. M. B. e Furuie, S. S. “Um Modelo de Autorização e Controle de Acesso para o Prontuário Eletrônico do Paciente em Ambientes Abertos e Distribuídos”, *Revista Brasileira de Engenharia Biomédica* 17, 3 (Setembro/Dezembro 2001), no prelo.
5. Motta, G. H. M. B.; Furuie, S. S.; Nardon, F. B.; Gutierrez, M. A. e Yamaguti, M. “Autorização e Controle de Acesso para o Prontuário Eletrônico do Paciente em Ambientes Abertos e Distribuídos: uma Proposta de Modelo e Arquitetura”, *Anais do WSeg'2001 - Workshop em Segurança de Sistemas Computacionais*, (Março 2001), 92-97.
6. Obelheiro, R. R.; Fraga, J. S. e Westphall, C. M. “Controle de Acesso Baseado em Papéis para o Modelo CORBA de Segurança”, *Anais do 19º Simpósio Brasileiro de Redes de Computadores*, maio de 2001.
7. Object Management Group. *CORBA Security Service Specification*. In: http://www.omg.org/cgi-bin/doc?formal/98-12-17.
8. Object Management Group. *Resource Access Decision Facility*. In: http://www.omg.org/cgi-bin/doc?dtc/00-08-06
9. Oh, S. e Park, S. “Enterprise Model as a Basis of Administration on Role-Based Access Control”, *Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications*, (2001), 150-158.
10. Osborn, S.; Sandhu, R. e Munawer, Q. “Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies”, *ACM Transactions on Information Systems Security* 3, 2 (Maio 2000), 85-106.
11. United States Department of Health and Human Services, Office of the Secretary. *Security and Electronic Signature Standards.* Federal Register, (Agosto 1998), 63:43241-43280.
12. Yeong, W.; Howes, T. e Kille, S. *Lightweight Directory Access Protocol (LDAP).* Internet Engineering Task Force – IETF, (Março 1995), In: http://www.ietf.org/rfc/rfc1777.txt?number=1777.