Modelo de Controle de Acesso para uma Arquitetura Orientada a Serviços Visando a Integração de Aplicações de Comando e Controle

Márcio Araújo Varchavsky, Eduardo Martins Guerra, Clóvis Torres Fernandes
Instituto Tecnológico de Aeronáutica, Pça Mal. do Ar Eduardo Gomes,50 - Vila das Acácias - São José dos Campos - SP

Resumo — **Em um ambiente orientado a serviços a natureza distribuída das aplicações dificulta a implantação de segurança. Nesse contexto, necessita-se de um padrão de controle de acesso que se adapte a essas necessidades. Este artigo irá analisar alguns padrões existentes no contexto de integração do aplicativos de comando e controle das forças armadas., Por fim, será proposto um modelo visando atender os principais requisitos da implantação da arquitetura orientada a serviços nesse contexto.**

Palavras-chaves — **Comando e controle, autenticação, controle de acesso, arquitetura orientada a serviços, SOA.**

I. INTRODUÇÃO

A arquitetura orientada a serviços (SOA) é um paradigma de arquitetura de software que define o uso de diversos serviços disponíveis numa rede para a implementação de regras de negócio. Os serviços podem ser de tecnologias e plataformas diversas, bastando que eles possuam uma interface padrão e uma comunicação feita por um protocolo bem definido.
O uso dessa arquitetura atende à maioria dos requisitos exigidos para a integração dos aplicativos de comando e controle das Forças. Uma arquitetura de integração com esse paradigma é a atual proposta para solução desse problema.
O uso da orientação a serviços leva, porém, à adição de alguns novos problemas, sobretudo na implantação de segurança. Em particular, há um acréscimo de complexidade nos aspectos de autenticação e controle de acesso, devido a sua distribuição e natureza não-centralizada.
Este artigo abordará as dificuldades para implementar a autenticação e o controle de acesso numa arquitetura orientada a serviços, além de propor uma solução para ser utilizada na arquitetura de integração.

II. DIFICULDADES DE SEGURANÇA EM UM AMBIENTE SOA

A arquitetura orientada a serviços tem sido usada para interoperar sistemas heterogêneos, independente de plataforma. A padronização da comunicação no ambiente que envolve os serviços propicia a elevada troca de informações necessária para a integração. Infelizmente, a disponibilidade de tal quantidade de informações e funcionalidades através de serviços necessita de controles de segurança adequados à arquitetura [1].
Um sistema tradicional possui os dados necessários para conhecer a priori cada usuário que o utiliza e o que cada um pode fazer. Um usuário também tem o conhecimento prévio de quais sistemas vai necessitar, já possuindo os dados necessários para cada autenticação, e com autorização prévia para realizar um conjunto de operações em cada um deles.
No exemplo da Figura 1, um usuário necessita inserir dados de um vôo, fazendo uso de três sistemas. Ele deve se autenticar em cada um deles para realizar as operações. Cada sistema deve manter a sua própria base de usuários e de autorizações, além de verificar na requisição se o usuário tem autorização para realizar a operação.

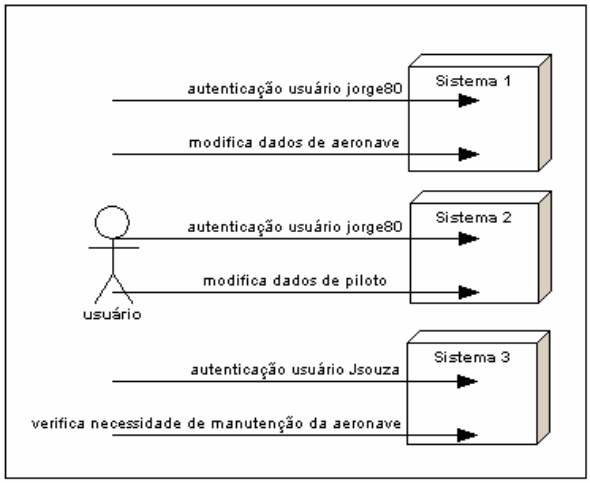

Fig. 1. Sistemas não integrados

No cenário de integração, o usuário iria apenas executar uma única operação de mais alto nível em um serviço intermediário, que trataria de identificar o fluxo de execução dos outros serviços, roteando as requisições e manipulando a resposta. Esse processo é chamado de orquestração de serviços [2].
A realização da mesma operação de inserir dados de um vôo, mas agora na arquitetura de integração, pode ser vista na figura 2. Já é possível perceber a necessidade da reformulação da autenticação e do controle de acesso., de forma a torna-la única e centralizada.
Nesse cenário, o usuário, para realizar uma operação, deveria ter conhecimento sobre todos os serviços que seriam utilizados, enviando juntamente com a requisição os dados de autenticação em cada serviço.
Uma atualização de regras de negócio onde ocorra uma troca de um dos serviços necessários para a composição da operação, que deveria ser transparente ao usuário final, acabaria não sendo. Numa troca, por exemplo, do serviço do sistema 3 por um serviço de um sistema 4, o usuário deveria saber que os dados de autenticação a serem enviados devem incluir os dados que foram cadastrados no sistema 4 e não mais os dados do sistema 3.

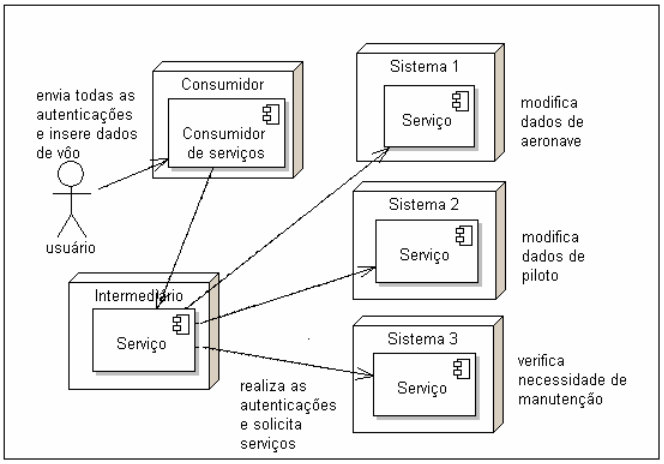

Fig. 2. Sistemas em arquitetura SOA mantendo autenticação e controle de acesso individuais

Além disso, o sistema 4 incluído deveria garantir o acesso adequado a todos os usuários que conseguiam realizar a operação antes da mudança, quando utilizavam o serviço do sistema 3. Caso contrário, usuários que antes realizavam com sucesso a operação de inserir dados de vôo não mais poderão fazê-lo.

Percebe-se também a dificuldade em se manter dados de autenticação e de autorização. Caso se queira mudar a senha de um usuário, ou apenas retirar sua autorização para inserir dados de vôo, serão necessárias mudanças nas bases de dados de todos os sistemas requisitados na operação.

O ambiente orientado a serviços, portanto, exige mecanismos de autenticação e controle de acesso que acompanhem o dinamismo e o fraco acoplamento propiciados pela arquitetura.

Na proposta de integração dos aplicativos de C2, a solução atual para evitar o envio de múltiplos dados de autenticação é a centralização dos dados de todos os usuários no servidor intermediário, que faria portanto a autenticação (única) dos usuários.

Este servidor possui também outras funcionalidades necessárias na integração, como roteamento, orquestração e tradução de serviços. Seu propósito é promover o desacoplamento entre as aplicações e coordenar a interação entre elas, fazendo o trabalho de um ESB [3] e mais alguma coisa. O intermediário poderá inclusive aparecer diversas vezes na arquitetura. Na figura 3 pode-se observar o intermediário e os seus componentes.

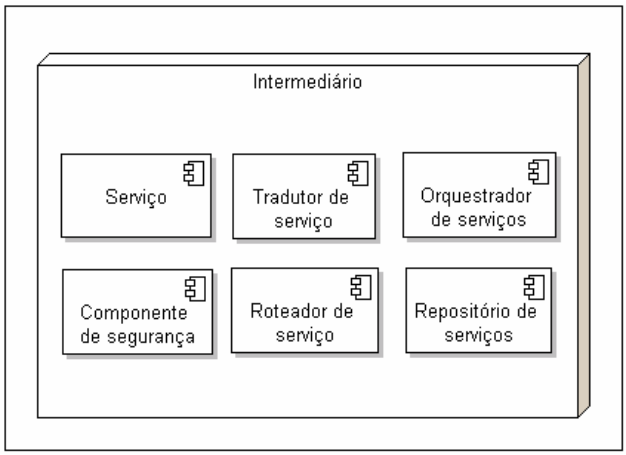

Fig. 3. Servidor intermediário usado na integração

Essa centralização de dados de autenticação, porém, é bastante perigosa, principalmente em aplicações militares. Apesar de propiciarem um processo único de autenticação, armazenam dados sigilosos em apenas um único ponto. Soma-se a esse problema a concentração elevada de usuários, originados dos sistemas de três forças armadas, num único registro.

A centralização também traz desvantagens para os dados de controle de acesso. Num ambiente heterogêneo de aplicações de três forças armadas, uma decisão centralizada de autorização retira a autonomia individual dos sistemas, com administrações próprias.

III. PADRÕES EXISTENTES PARA SEGURANÇA EM UM AMBIENTE SOA

Uma técnica recomendada para implantar segurança num ambiente SOA é a implementação de um serviço de segurança [4], como pode ser visto na Fig. 4. As diversas funcionalidades de segurança como autenticação, autorização, criptografia, assinaturas, entre outras, estariam presentes. Alguns serviços podem aparecer mais vezes, evitando assim centralizações prejudiciais. A abordagem retira a presença de implementações de segurança diferentes em cada serviço, aumentando a confiança no sistema como um todo. O serviço também facilitaria o controle de informações de autenticação e autorização de usuários, centralizando as bases de dados e as tomadas de decisões.

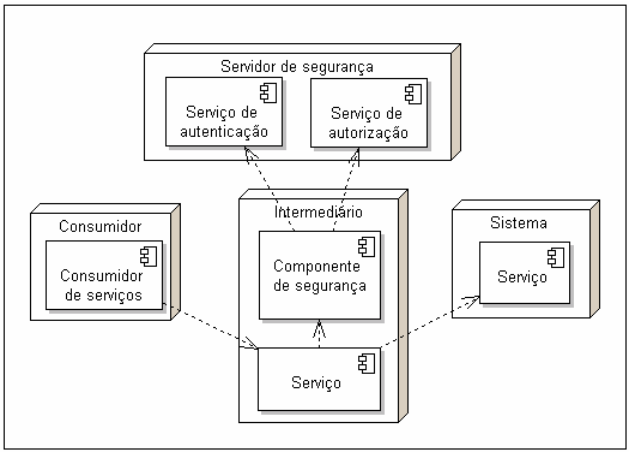

Fig. 4. Arquitetura de um serviço de segurança

Para viabilizar este tipo de serviço, é necessário definir como será a troca de informações de segurança no padrão de troca mensagens utilizado. O padrão *WS-Security* [5], para *Web Services*, especifica três mecanismos: integridade de mensagens, confidencialidade de mensagens, e habilidade de inserir *tokens* de segurança como parte da mensagem.

Alguns dos *tokens* que podem ser utilizados vêm de dois padrões, *Security Assertions Markup Language* (SAML) e *eXtensible Access Control Markup Language* (XACML).

O padrão SAML [6] especifica a semântica para a troca de algumas informações de segurança, como atributos, dados de autenticação e decisões de autorização. O XACML[7] define, entre outras coisas, uma linguagem para expressar políticas de segurança, tomar decisões de autorização baseado nessas políticas e semântica para a troca de informações relacionadas à autorização.

O uso destes padrões viabiliza a utilização de um serviço de segurança na arquitetura de integração. Apesar dos padrões definirem o formato e a manipulação de dados, eles acomodam o uso de diversos modelos e tecnologias de segurança. A análise para escolha de um padrão de controle de acesso adequada será feita na próxima seção.

IV. PADRÕES DE CONTROLE DE ACESSO

Um padrão de controle acesso pode ser descrito formalmente [8] usando as noções de usuários, sujeitos, objetos, operações e permissões, além das relações entre eles. Um usuário é aquele que está diante da máquina, possuindo um ou mais identificadores que podem estar simultaneamente ativos. Um processo agindo de acordo com o usuário é um sujeito. Um objeto é um recurso acessível num sistema. Uma operação é a manifestação da atuação de um sujeito, geralmente em algum objeto. E a permissão é a autorização para que uma operação possa ser feita num determinado objeto.

Um padrão de controle de acessos muito usado atualmente é o *Role Based Access Control* (RBAC). O grande feito deste padrão foi inserir um intermediário nas associações diretas que eram feitas antes entre usuários e permissões. O RBAC associa papéis às permissões e um papel deve ser explicitamente designado a um grupo de usuários. Esses papéis geralmente correspondem aos papéis reais desempenhados por cada usuário na organização. Esta forma de associação reduz substancialmente o número de associações a serem mantidas no sistema.

Um exemplo seria um usuário solicitar a permissão de leitura de um determinado dado. O usuário estaria associado a um papel "Chefia", e a esse papel estaria ligada a permissão de leitura aquele dado. Dessa forma, o usuário estaria autorizado a ler tal dado. Um diagrama representando os relacionamentos do RBAC pode ser visto no exemplo da figura 5.

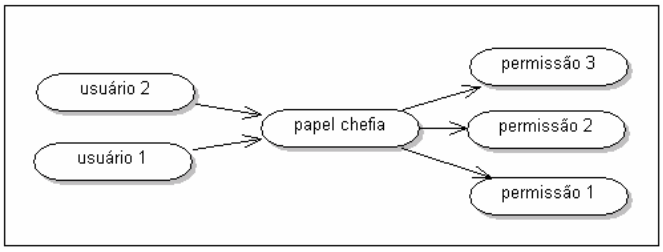

Fig. 5. Diagrama representando RBAC

O uso do padrão facilita a consulta da decisão e manutenção dos dados em relação a padrões que associam diretamente o usuário à permissão.

Identificar se o usuário pode realizar a operação de dados sigilosos exigiria a consulta de todas as permissões dos usuários, ou de todos os usuários associados ao objeto com essa operação. Para o RBAC, basta identificar os papéis do usuário e realizar uma consulta de permissões em seus papeis, verificando se algum deles possui tal acesso.

Remover esta permissão do usuário, sem o RBAC, necessitaria a remoção de todas as associações usuário-permissão. No RBAC seria necessário retirá-lo do papel que desempenha, fato que provavelmente acorreu também na organização.

O controle de acessos baseado em atributos, ABAC [9], define permissões se baseando em características relevantes de segurança, conhecidas como atributos. O padrão define três tipos de atributos: atributos do sujeito, atributos do objeto e atributos do ambiente. Os dois primeiros são características de cada um. O último representa um grande avanço, pois foi um fator ignorado na maioria dos padrões anteriores, e descreve o contexto em que o acesso é realizado.

A decisão da autorização é feita consultando uma ou mais regras pré-definidas, que admitem como entrada os valores resultantes da atribuição de atributos no sujeito, no objeto e no ambiente envolvidos, retornando uma booleana de decisão.

Exemplos de atributos seriam: Posto(.), Cargo(.), atributos pré definidos para o sujeito; Sigilo(.), atributo para o objeto e Data(.), atributo para o ambiente. O resultado da aplicação dos atributos poderia ser:

Posto(s) = "Major"
Cargo(s) = "Chefia"
Sigilo(o) = "Secreto"
Data(a) = "12/10/2007"

E uma regra:

Pode_acessar(s,o,a) = (Cargo(s) = "Chefia") ^
(Sigilo(o) = "Secreto")

A booleana resultante da aplicação de todas as regras resulta na decisão de acesso.

Num ambiente orientado a serviços, com diversas integrações sendo interoperadas, o número de papéis possíveis pode crescer significativamente, além da possibilidade de haver usuários em vários papéis. Esse fato insere no controle de acesso as dificuldades de manutenção do uso de associação direta usuário permissão. Já o ABAC, por suas características de granularidade fina semelhantes às da orientação a serviços, se adequa muito melhor.

Além disso, o ABAC se comporta de forma mais eficiente em ambientes dinâmicos por conta dos atributos temporais do ambiente. Inserir uma regra de acesso que permita uma operação apenas para militares de serviço, no RBAC, exigiria a criação de papéis complexos e uma atualização constante deles. No ABAC, apenas exigiria a inserção na regra atual de uma cláusula avaliando um atributo temporal relacionado ao período de serviço do militar.

O padrão XACML [7] tem total suporte para o ABAC, trazendo métodos para tomada de decisão baseada nos três tipos de atributos, inclusive com padrões para expressar as regras decisão.

O uso do padrão é considerado portanto o mais adequado no ambiente SOA.

V. DEFINIÇÃO DE UM SERVIÇO DE AUTENTICAÇÃO E AUTORIZAÇÃO

Os serviços de segurança comumente implementados possuem a capacidade de tomar as decisões de autenticação e autorização de todo o sistema, de forma centralizada. Um componente de autenticação com os dados de todos os usuários suporta uma autenticação única no sistema. Um componente de controle de acesso toma a decisão de

autorização buscando dados em outros componentes, muitas vezes em outros serviços.

Em uma outra abordagem é utilizado um controle centralizado de autenticação e autorização, que coleta diversas decisões de autorização e toma uma decisão final, que é então anexada à requisição do cliente e enviada ao serviço [10].

A centralização dos dados de autenticação é comum mas implicam no perigo do armazenamento elevado de dados num único ponto, ainda mais sendo esses dados de natureza sigilosa.

Em um ambiente de integração, os aplicativos podem exigir uma autonomia de decisão por terem administrações próprias. Os aplicativos de comando e controle a serem integrados são geridos pelas três forças armadas, e muitos estão em diferentes setores da mesma força. Assim há a necessidade de uma estratégia não invasiva de tomada de decisão mas que não possua as implicações da decisão totalmente descentralizada vista no início do artigo.

O padrão escolhido de controle de acesso, o ABAC, propicia que a tomada de decisão de autorização, centralizada ou não, se faça baseando-se em informações (no caso os atributos) colhidas em diversos pontos do sistema.

Em [11] e [9] o cliente, ao solicitar acesso ao serviço, recebe a mensagem de que deve apresentar uma credencial de autenticação e uma credencial de autorização. A credencial de autorização é dada por uma entidade, que toma a decisão com base em atributos vindos dos diversos serviços integrados.

A arquitetura proposta será complementar à arquitetura desses dois trabalhos. A infra-estrutura da manutenção de dados de autenticação será mais detalhada, e a autoridade de quem decide a autorização será reforçada usando a divisão do sistema em domínios. Além disso, adaptações serão feitas já que no sistema abordado neste artigo os acessos aos serviços são feitos sempre através de um serviço intermediário.

A proposta de serviço de autenticação e autorização pode ser vista na figura 6.

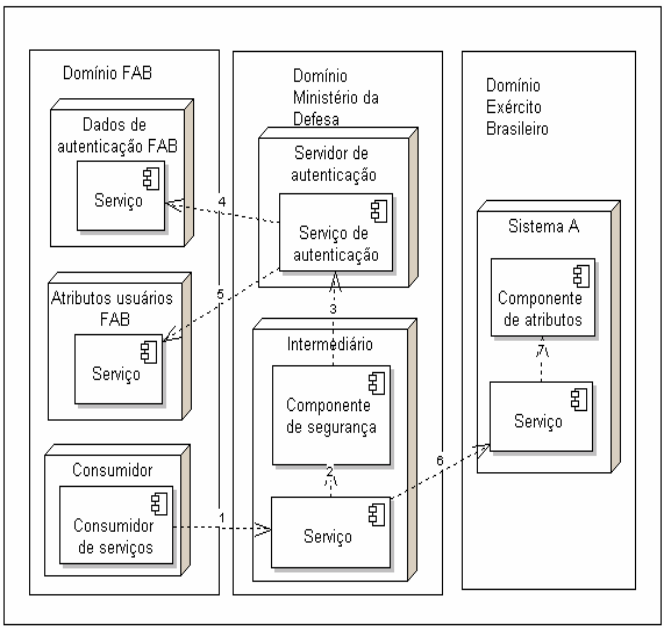

Fig. 6. Arquitetura de autenticação e autorização proposta

Ao acessar o sistema de integração, o usuário envia a requisição ao serviço intermediário em conjunto com os dados de autenticação. O Intermediário então toma a decisão de autenticação consultando um serviço de segurança. Esses serviços de segurança podem ser distribuídos, havendo vários locais onde os atributos do usuário podem ser buscados.

Uma diferença dos padrões comumente vistos é o serviço de autenticação roteando o processo para um serviço de autenticação específico de um determinado domínio, que toma a decisão. O serviço do intermediário poderia, por exemplo, identificar que determinado usuário pertence ao domínio do Exército e buscar a decisão de autenticação no serviço do próprio domínio.

A gerência desse domínio possui autoridade sobre os dados dos seus usuários, e informa a decisão de autenticação. A escolha do padrão de autenticação também fica a cargo do domínio e pode ser modificada de acordo com mudanças de necessidades, sendo essa modificação facilitada pelo isolamento da funcionalidade.

Portanto a centralização da gerência da autenticação propicia que o processo seja feito uma única vez. E os dados descentralizados facilitam a manutenção dos usuários e permitem a gerência própria das informações.

O serviço de autenticação busca também num serviço do domínio os atributos do usuário. Ambas as mensagens, de autenticação e de atributos, são assinadas e atestam a veracidade e a origem das informações a serem lidas por qualquer outro domínio. Elas serão anexadas à requisição que o serviço intermediário enviará para o serviço alvo. Atributos de ambiente também podem ser adicionados à mensagem, considerados no próprio domínio da análise,ou obtidos consultando algum outro serviço.

O serviço que recebe a mensagem possui então a tarefa de, antes de atender à mensagem, tomar a decisão de autorização. Os atributos recebidos do usuário são adicionados a mais atributos de ambiente e aos atributos do recurso, usando algum componente presente no próprio serviço. Esses atributos estão no escopo da aplicação, e são mantidos pela autoridade que gerencia o sistema. Com base nestes atributos, a decisão é então tomada.

As decisões de autorização são descentralizadas e dão autonomia aos sistemas, mas são feitas com base nos atributos e não no usuário. Apesar do sistema ter a visibilidade de quem está fazendo a requisição, ele não precisa manter um controle de acessos baseado no usuário, mas apenas nas suas características providas pelo servidor de segurança.

As trocas de mensagens entre domínios terão sempre sua veracidade atestada por assinaturas. Pode-se criar algum serviço, no escopo do domínio, que isole a funcionalidade de assinatura e de verificação da mesma.

A manutenção dos dados de autenticação e autorização em relação ao usuário é feita uma única vez no serviço de segurança. Em relação à autorização, é necessário apenas modificar alguns atributos do usuário à medida que seu papel no sistema modifica. Nos serviços, a manutenção é feita modificando atributos de ambiente, atributos do recurso e as regras de tomada de decisão.

A parte da decisão de autorização que cabe ao serviço pode também ser isolada num outro serviço de segurança, mas este sendo administrado pelos mesmos gestores do serviço. Assim, a autonomia dos gestores sobre o serviço é garantida.

VI. CONCLUSÃO

No artigo foi demonstrada a dificuldade em se implantar uma infra-estrutura de autenticação e controle de acesso num sistema que integra outros sistemas usando a arquitetura orientada a serviços. Esse é o caso da proposta de integração dos aplicativos de comando e controle das forças armadas.

Foi feito o uso de um padrão comum de arquitetura de segurança em ambiente orientado a serviços, que é a utilização de um serviço que propicie as funcionalidades de autenticação e autorização. A padronização da troca de mensagens necessária para o uso do serviço de segurança é viabilizada por alguns padrões existentes.

Uma análise foi feita e o padrão de controle de acesso considerado mais apropriado num ambiente orientado a serviços foi o controle de acesso baseado em atributos. Sua estrutura permite maior dinamismo na tomada de decisão de autorização e permite também esta decisão seja feita de forma mais descentralizada com base em informações coletadas em diversos componentes do sistema.

O artigo definiu uma arquitetura de autenticação e autorização a ser utilizada no sistema de integração dos aplicativos de comando e controle. O controle da autenticação será feito de forma centralizada no serviço de segurança, porém com base em decisões do domínio onde se encontra o usuário.

A decisão de autorização proposta será feita com base nos atributos coletados em alguns componentes, e as regras de decisão com base nos atributos são mantidas pelas autoridades que gerenciam o aplicativo.

Algumas das principais contribuições do artigo foram: adaptação da arquitetura de serviço de segurança para o ambiente com o uso do serviço intermediário; distribuição das coletas de dados e das decisões de autenticação e autorização nos domínios independentes a serem integrados para garantir a autonomia dos participantes.

REFERÊNCIAS

[1] B. French, *"Implementing SOA Security", Governance requirements for SOA security,* IBM, 2006.

[2] OASIS, "*Web Services Business Process Execution Language v2.0* (WSBPEL)", OASIS *Homepage*, http://docs.oasis-open.org/wsbpel/v2.0/

[3] M Keen, A Acharya, S Bishop, A Hopkins, S Milinski, *Patterns: Implementing an SOA Using an Enterprise Service Bus*, IBM Redbooks, July, 2004.

[4] R. Kanneganti, P. Chodavarapu, "*SOA Security In Action*", 1st ed, vol 1, Ed Manning, 2007, pp. 26-28

[5] OASIS, "*Web Services Security* (WSS)", OASIS *Homepage*, http://docs.oasis-open.org/wss/v1.1/

[6] OASIS, *The Security Assertions Markup Language* (SAML) OASIS *Homepage*, http://docs.oasis-open.org/security/saml/v2.0/

[7] OASIS, *The eXtensible Access Control Markup Language* (XACML) OASIS *Homepage*, http://docs.oasis-open.org/xacml/2.0/

[8] D. F. Ferraiolo, D. R. Kuhn, R. Chandramouli, "*Role Based Access Control*", 1st ed, vol 1, Ed Artech House, 2003, pp. 4, 5, 9-12, 51-59

[9] E.Yuan, J. Tong, "*Attributed Based Access Control (ABAC) for Web Services*", *Proceedings of the IEEE International Conference on Web Services, IEEE Computer Society,* 2005.

[10] S. Indrakanti, V. Varadharajan, "*An Authorization Architecture for Web Services*", *Information and Networked Systems Security Research, Department of Computing, Macquarie University*, 2006.

[11] R. Erber, C. Schläger, G. Pernul, "*Patterns for Authentication and Authorisation Infrastructure" International Workshop on Secure Systems Methodologies using Patterns*, 2007

M. A. Varchavsky, marciovar@gmail.com, Tel +55-12-39450366; E. M. Guerra, guerraem@gmail.com, Tel +55-12-39473776; C. T. Fernandes clovistf@uol.com.br, Tel. +55-12-39475983