Um Modelo de Composição de Políticas de Qualidade de Proteção para Serviços *Web* Compostos

Davi da Silva Böger[1] **, Michelle S. Wangham**[2] **, Joni Fraga**[1*]**, Paulo Mafra**[1*]

[1]Departamento de Automação e Sistemas (DAS)
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

[2]Grupo de Sistemas Embarcados e Distribuídos–GSED/CTTMAR
Universidade do Vale do Itajaí (UNIVALI) – São José – SC – Brasil

{dsboger,fraga,mafra}@das.ufsc.br, wangham@univali.br

Abstract. *The description languages for Web Services composition deal specifically with the definition of the business process logic and do not provide support for security aspects regarding the involved Web Services. This paper uses WS-Policy, WS-BPEL and WSCDL standards to propose a model to check, in a preliminary way, the compatibility of the quality of protection policies of the business process participants and builds the composite process policy.*

Resumo. *As linguagens de descrição de composição de Serviços* Web *tratam especificamente da definição da lógica dos processos de negócio e não dão suporte a especificação dos aspectos de segurança dos Serviços* Web *envolvidos. Este artigo define um modelo que, com o auxílio do padrão* WS-Policy *e das linguagens WS-BPEL e WS-CDL, promove a verificação preliminar da compatibilidade das políticas de qualidade de proteção dos participantes de um processo de negócio e constrói a política composta do processo.*

1. Introdução

A Internet provê uma poderosa infra-estrutura de comunicação e, devido a isto, a realização de negócios que usufruem desta cresceu muito nos últimos anos, significando um aumento no número de aplicações distribuídas que ultrapassam as fronteiras de uma única organização. Diante da necessidade de interação entre as aplicações de diferentes organizações, que geralmente não foram desenvolvidas para serem interoperáveis, uma nova caracterização de sistemas distribuídos surgiu possibilitando assim a troca de informações em sistemas heterogêneos - os **Serviços *Web*** [WS-ARCHITECTURE 2004].

Os Serviços *Web* seguem uma arquitetura orientada a serviços (SOA) e são componentes de *software* projetados para suportar interações sobre a Internet. As principais características que os tornam uma tecnologia promissora são [WS-ARCHITECTURE 2004]: (1) possuem um modelo fracamente acoplado e transparente que garante a interoperabilidade entre os serviços, sem que estes necessitem ter o conhecimento prévio de quais tecnologias estão presentes em cada lado da comunicação; (2) são auto-contidos e auto-descritivos, (3) usam padrões abertos amplamente aceitos como o HTTP e o XML e (4) tornam mais fácil a composição de diferentes provedores, visando formar serviços mais complexos e sofisticados. Um Serviço *Web* representa uma função de negócio completa e pode ser usado individualmente ou pode fazer parte

*Bolsista CNPq.

de uma composição envolvendo outros serviços, o que torna essa tecnologia interessante para empresas que desejam fazer negócios de forma dinâmica.

Devido à necessidade de uma rápida adaptação dos negócios para atender novas condições de mercado, cada vez mais as organizações procuram se unir para criar processos de negócios, que descrevem serviços complexos, que transpassam os seus limites organizacionais e que são providos por diferentes parceiros. No entanto, apesar da tendência para trabalhos colaborativos, muitas organizações ainda têm receios em compartilhar informações sensíveis, quando há a necessidade de colaboração com parceiros desconhecidos [WANGHAM et al. 2005]. Ainda mais em um sistema aberto, como a Internet, onde questões relacionadas à segurança são de fundamental importância para a realização de processos de negócio.

Sempre com o objetivo de manter a interoperabilidade, os Serviços *Web* se baseiam em padrões para solucionar questões de segurança. Há padrões para prover segurança às trocas de mensagens, tais como o [WS-SECURITY 2006], para estabelecer relações e criar domínios de confiança [WS-TRUST 2007, WS-FEDERATION 2006] e para expressar requisitos de segurança [WS-POLICY 2007, WS-SECURITYPOLICY 2007]. Entretanto, quando se considera não mais um serviço único, mas sim uma composição de Serviços *Web*, constata-se que a segurança dos processos de negócios não foi ainda profundamente investigada e que ainda não existem soluções concretas e completas [CHARFI e MEZINI 2005, CARMINATI et al. 2005].

Em uma composição de Serviços *Web*, cada organização definirá um conjunto de características de segurança para os Serviços *Web* que irá disponibilizar e diferentes organizações podem ou não concordar, em face das restrições impostas, em trocar informações. Para que a colaboração possa atingir seu objetivo como função de negócio, é preciso que as organizações envolvidas aceitem as restrições impostas umas às outras. Nos Serviços *Web*, diversos tipos de restrições de segurança podem ser descritos na forma de políticas *WS-Policy*[1] [WS-POLICY 2007] e, uma vez que todas as partes envolvidas em uma colaboração tenham descrito suas restrições nesse formato, é possível verificar se há incompatibilidades que venham a impedir o cumprimento dos objetivos da colaboração.

Este artigo tem por objetivo descrever um modelo que realiza a verificação preliminar das restrições de segurança dos participantes de uma colaboração. Essa verificação está baseada na noção de compatibilidade de políticas [WS-POLICY 2007] e tem como resultado a política composta da colaboração. A política composta expressa o que há de comum nas restrições de segurança dos participantes e revela os pontos onde não há compatibilidade. Os participantes podem se basear nessas informações para assegurar que a colaboração não será interrompida por restrições de segurança. Por resultar na política composta, a verificação dos requisitos de segurança será denominada de **composição de políticas de qualidade de proteção**. De forma a comprovar a sua aplicabilidade, um protótipo do modelo foi implementado e integrado a uma composição de Serviços *Web*.

O restante do artigo está estruturado da seguinte forma: a seção 2 revisa alguns conceitos de composição de Serviço *Web* e levanta questões relacionadas a segurança;

[1] O padrão WS-Policy não abrange todas as características de segurança, pois limita-se à especificação de controles de confidencialidade, integridade e autenticação de mensagens, não abrangendo controle de acesso. Os mecanismos descritos nessas políticas dizem respeito à proteção das mensagens, e por isso estas serão chamadas de **políticas de qualidade de proteção**.

a seção 3 apresenta o modelo de composição de políticas proposto; a seção 4 descreve o protótipo implementado do modelo; a seção 5 apresenta alguns trabalhos de escopo semelhante; a seção 6 conclui o artigo.

2. Segurança em Serviços *Web* Compostos

Conforme definido em [PELTZ 2003], a orquestração e a coreografia de Serviços *Web* são abordagens baseadas em padrões abertos para conectar os serviços e criar processos de negócios de alto nível. A orquestração apresenta um processo de negócio executável, que pode interagir com serviços internos e externos à organização, que descreve como os Serviços *Web* podem interagir em nível de mensagem e que inclui a lógica de negócio e a ordem de execução das tarefas (atividades). Com a orquestração, o processo é sempre controlado por uma das partes do negócio, o que difere da coreografia que é mais colaborativa e permite que cada parte envolvida descreva sua participação na interação. A coreografia descreve tipicamente as trocas de mensagem públicas que ocorrem entre os Serviços *Web*, ao invés do processo de negócio específico que uma única parte executa [PELTZ 2003].

Tratando-se da composição de Serviços *Web*, já existem especificações que padronizam a forma de descrever processos de negócio. Conforme [CHARFI e MEZINI 2005], o padrão mais popular e de grande aceitação na academia e na indústria é a Linguagem para Execução de Processos de Negócio de Serviços *Web* (*Web Services Business Process Execution Language*) [WS-BPEL 2007]. Essa linguagem serve para descrever um processo de negócio executável, sob o ponto de vista de uma única organização. A especificação de um processo WS-BPEL é divida em duas partes: uma parte declara os tipos de serviços envolvidos na orquestração, as variáveis, tratadores de falhas e eventos e outros recursos a serem usados no contexto do processo; a outra parte declara o *workflow* por meio de atividades. Os serviços parceiros são declarados usando o conceito de *partnerLink*, que descreve uma relação entre um participante e o orquestrador e as interfaces que cada um implementa. Para permitir a comunicação com Serviços *Web* parceiros, a WS-BPEL define as atividades *invoke*, *receive*, *reply* e *pick* e os tratadores de eventos.

Para construir coreografias, o padrão (candidato) W3C é a Linguagem de Descrição de Coreografia de Serviços *Web* (*Web Services Coreography Description Language*) [WS-CDL 2005]. Uma descrição WS-CDL é um conjunto de definições contidas dentro de um elemento `<package>`. Essas definições podem especificar tipos de informação que serão refenciados nas coreografias, participantes, canais de comunicação e coreografias. Em uma coreografia são declaradas as variáveis que representam estado observável dos participantes, sub-coreografias, o fluxo principal de atividades e os tratadores de exceção e de finalização. Os participantes são declarados na forma de *roleTypes* que descrevem as interfaces que o participante implementa. Os canais de comunicação (*channelTypes*) indicam os pontos de acesso aos participantes. A WS-CDL possui apenas uma atividade para representar a comunicação entre os participantes, a atividade *interaction*, que descreve uma seqüência de trocas de mensagens sobre um canal de comunicação provido por um dos dois serviços envolvidos.

Na composição de Serviços *Web*, as aplicações tornam-se ainda mais visíveis, expondo suas funcionalidades, seus fluxos de negócios, processos, políticas e arquiteturas internas. A colaboração entre diferentes parceiros envolve o atravessamento de várias camadas de segurança. Como os limites administrativos precisam ser transpassados, os

processos de negócios estarão sob diversos modelos administrativos e também sob diversos mecanismos e tecnologias de segurança. Por exemplo, um provedor de serviço pode impor determinadas restrições sobre os requerentes de seu serviço, tais como a exigência de um determinado mecanismo de autenticação ou de um algoritmo de criptografia para proteger as mensagens, ou ainda exigir que as credenciais dos parceiros sejam emitidas por uma entidade que este confie. Neste cenário, uma política de qualidade de proteção especifica quais mecanismos de autenticação, algoritmos de cifragem e de assinatura digital são suportados por um Serviço *Web* parceiro. Pode ocorrer, no entanto, que os requisitos de dois participantes do processo não sejam compatíveis quanto aos mecanismos de proteção e, nesse caso, se o processo chegar ao ponto em que esses dois participantes precisam interagir, uma falha ocorrerá no processo, pondo em risco toda a colaboração. É possível, porém, verificar com antecedência se há concordância entre as políticas de qualidade de proteção dos participantes. Logo, é possível saber se há risco de o processo de negócio não ser finalizado corretamente por uma incompatibilidade de requisitos de proteção. O modelo proposto neste trabalho se ocupa desta problemática.

O padrão *WS-Policy* [WS-POLICY 2007] permite expressar requisitos não-funcionais, como mecanismos de proteção, de Serviços *Web* na forma de políticas. A unidade básica para expressar um requisito é uma **asserção de política**, que possui semântica específica de domínio. Asserções podem ser combinadas em **alternativas de política**. Cada alternativa define um conjunto de requisitos que devem ser satisfeitos em conjunto para satisfazer a alternativa. Uma **política**, por sua vez, é uma coleção de alternativas de política. Satisfazer a política significa satisfazer uma dessas alternativas.

A especificação *WS-SecurityPolicy* [WS-SECURITYPOLICY 2007] define um conjunto de asserções para serem usadas em políticas *WS-Policy* que são relativas ao uso dos padrões [WS-SECURITY 2006], [WS-TRUST 2007] e [WS-SECURECONVERSATION 2007] de segurança para Serviços *Web*.

A intersecção de políticas, operação importante para o modelo proposto, é definida na *WS-Policy*. Tal operação consiste em construir, a partir de duas políticas, uma terceira que contenha as alternativas compatíveis de ambas políticas interseccionadas. Duas alternativas são compatíveis se todas as asserções de uma alternativa forem compatíveis com ao menos uma asserção da outra. A compatibilidade entre asserções é específica para cada domínio, no entanto a *WS-Policy* recomenda que, sempre que possível, os nomes dos elementos XML que representam as asserções sejam suficientes para a verificação de compatibilidade.

3. Modelo de Composição de Políticas de Qualidade de Proteção

No modelo proposto neste trabalho, chama-se de composição de políticas de qualidade de proteção a verificação da concordância entre as políticas dos participantes, pois esta terá como resultado um conjunto de políticas associadas às trocas de mensagens do processo de negócio e esse conjunto pode ser entendido como a política composta da colaboração. A política composta indica quais os mecanismos de proteção que devem ser usados nas trocas de mensagem do processo e também demonstra se existe algum ponto onde não há compatibilidade dos mecanismos requeridos pelos participantes do processo.

Visando oferecer interoperabilidade, requisito necessário para o funcionamento de processos de negócios em sistemas abertos, o modelo proposto está baseado nas principais especificações relacionadas a Serviços *Web* e a composição de serviços. As-

sim sendo, as políticas devem ser descritas e anexadas de acordo com a *WS-Policy* e *WS-PolicyAttachment*, as asserções de segurança devem estar de acordo com a *WS-SecurityPolicy* e os processos de negócio podem ser descritos em WS-BPEL ou WS-CDL.

A Figura 1 apresenta uma visão geral do modelo proposto. A composição das políticas é feita por um Serviço *Web*, chamado de **serviço compositor**, responsável por receber a descrição do processo de negócio de um cliente (passo 1), por obter as descrições WSDL e as políticas *WS-Policy* de cada participante (passo 2) e, por fim, por analisar as informações dos participantes em conjunto com a descrição do processo para gerar a política composta (passo 3). Esta é formada de um conjunto de políticas *WS-Policy* anexadas às trocas de mensagem da colaboração que indicam quais mecanismos de proteção podem ser usados em cada mensagem trocada.

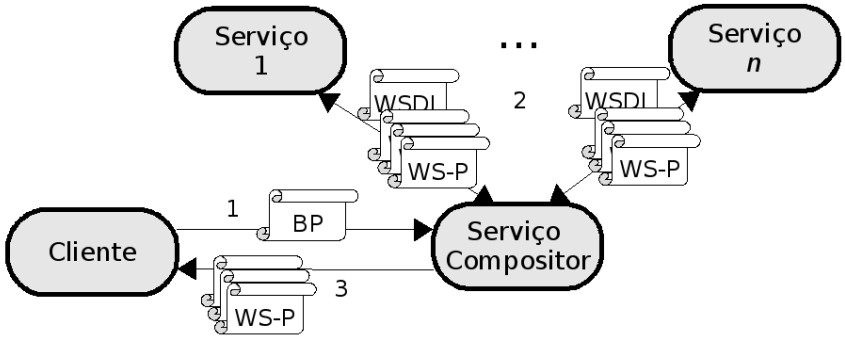

Figura 1. Visão Geral do Serviço Compositor de Políticas

Como será evidenciado nas seções seguintes, há etapas do serviço compositor de políticas que são dependentes do formato da descrição do processo de negócio. O modelo proposto prevê a utilização dos padrões atuais de descrição, logo será descrito como o serviço compositor deve proceder com orquestrações em WS-BPEL e coreografias em WS-CDL[2].

A Figura 2 ilustra, de forma resumida, as atividades realizadas pelo serviço compositor. Essas atividades serão detalhadas nas seções seguintes.

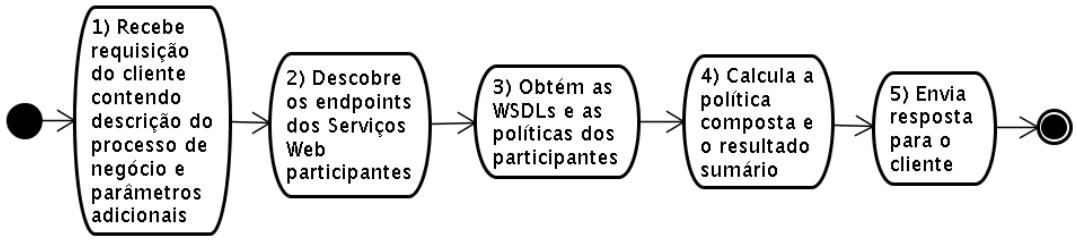

Figura 2. Atividades Realizadas pelo Serviço Compositor

3.1. Descoberta dos Participantes da Composição

Primeiramente, o serviço compositor recebe a descrição do processo de negócio do cliente em WS-BPEL ou WS-CDL. Essa descrição pode vir acompanhada do conjunto dos participantes do processo. Caso não venha, o serviço compositor deve obter a lista dos participantes por meio de uma análise da descrição do processo de negócio. Cada tipo de descrição possui um conceito ou conjunto de conceitos para representar um Serviço

[2]O modelo, porém, não está restrito a esses formatos e pode ser aplicado a qualquer tipo de processo de negócio, desde que obedeça à premissa de ser redutível a um conjunto de mensagens enviadas de um participante a outro.

Web participante, mas o serviço compositor trata, nesta etapa, apenas dos *endpoint references* (EPRs)[3] [WS-ADDRESSING 2006] desses serviços, pois as políticas de segurança descritas segundo a *WS-SecurityPolicy* devem sempre estar associadas aos elementos da WSDL relacionados à implementação do serviço, como *endpoint* e *binding* [WS-SECURITYPOLICY 2007].

Existem algumas dificuldades com relação à obtenção dos EPRs a partir da descrição do processo, já que além de ser dependente do tipo de processo e do formato da descrição, a maioria das linguagens de descrição permite que o *endpoint* seja resolvido dinamicamente, com base em informações que só estarão disponíveis durante a execução do processo. Esses participantes, descobertos dinamicamente, são tratados de forma diferente pelo serviço compositor, como será descrito na seção 3.3.

Em um processo WS-BPEL, há dois tipos de participantes: um é o serviço orquestrador, implicitamente definido como o serviço que executa o *workflow*; o outro são aqueles identificados pelos *partnerLinks*. O serviço orquestrador não é identificado na orquestração, logo seu EPR precisa ser declarado em um parâmetro externo à descrição do processo. Quanto aos *partnerLinks*, há duas formas de se associar um EPR a estes [WS-BPEL 2007]: em uma atividade *assign* no *workflow* do processo ou usando um mecanismo externo ao processo, previsto na especificação.

Nas coreografias WS-CDL, os participantes são representados por meio de *roleTypes*. A correspondência entre o *endpoint* e o *roleType* é declarada no canal de comunicação que deve ser usado para trocar mensagens com o *endpoint*. Cada canal é representado por um *channelType* que possui um elemento `<reference>`. Este indica o tipo de informação que deve ser usada como referência para o *endpoint*. Para que um tipo de canal seja usado nas comunicações da coreografia, é necessário que seja declarada uma variável do tipo de canal desejado e essa variável irá conter as informações necessárias para interagir com o *endpoint*. No entanto, a WS-CDL não restringe os meios de se atribuir valores a essas variáveis. Neste modelo, foi usado um parâmetro adicional contendo um mapeamento entre *roleTypes* e EPRs.

3.2. Descoberta das Interfaces WSDL e das Políticas WS-Policy

Após a descoberta dos EPRs, o serviço compositor poderá partir para a obtenção das descrições WSDL e das políticas *WS-Policy* dos participantes. Há diversas formas para obter essas informações e este modelo prevê o uso das seguintes, na respectiva ordem de precedência:

1. o serviço compositor invoca as operações *Get* ou *GetMetadata* implementadas pelo *endpoint* do participante [WS-METADATAEXCHANGE 2006];
2. o cliente passa ao serviço compositor, na invocação, URIs que apontam para a descrição WSDL que contém referências para as políticas do participante;
3. o EPR do participante, descoberto pelo serviço compositor, contém um elemento `<wsa:Metadata>` [WS-ADDRESSING 2006] que contém os metadados do participante.

Se alguma descrição WSDL não for encontrada ou alguma das políticas *WS-Policy* referenciadas na WSDL de algum dos participantes não for encontrada, o serviço compositor de políticas irá emitir uma mensagem de erro e irá interromper a composição das políticas.

[3]Um *endpoint* é uma entidade física que implementa um Serviço *Web*. Um *endpoint reference* identifica unicamente um *endpoint* e para isso normalmente é usado um URI.

3.3. Composição das Políticas de Qualidade de Proteção

Após obter as informações dos serviços participantes da colaboração, o serviço compositor inicia a etapa de análise dessas informações. A análise está baseada na redução do processo de negócio a um conjunto de trocas de mensagens. A verificação de cada envio de mensagem se dá pelo cálculo da intersecção das políticas de qualidade de proteção pertinentes das duas partes envolvidas.

As mensagens trocadas em uma interação devem fazer parte de uma operação descrita na WSDL do serviço provedor, e por isso este tem a possibilidade de anexar uma política de qualidade de proteção específica para cada mensagem pertencente à operação. A parte que invoca a operação (cliente), por outro lado, não pode anexar políticas específicas às mensagens descritas na WSDL do provedor. Isso não se faz necessário quando o próprio cliente é responsável por verificar se aceita as restrições impostas pela política do provedor, porém, no modelo proposto neste trabalho, essa verificação é delegada ao serviço compositor e é preciso estabelecer uma forma do cliente declarar suas restrições, se assim desejar.

A solução adotada no modelo proposto foi criar um novo sujeito de política [WS-POLICY 2007], chamado de **participante cliente**, que representa uma participação de um *endpoint* como cliente em trocas de mensagens de um determinado processo de negócio. Se um participante do processo anexar uma política a esse sujeito (como mostrado na Figura 3), estará declarando restrições às trocas de mensagem iniciadas por esse participante no âmbito do processo. Todas as asserções de política aplicáveis ao sujeito *endpoint* podem ser anexadas ao participante cliente. Essa é uma proposta de extensão do padrão *WS-PolicyAttachment* que permite maior flexibilidade no casamento entre os padrões para expressar políticas e as colaborações de Serviços *Web*.

```
<wsp:PolicyAttachment xmlns:wsp="...">
  <wsp:AppliesTo>
    <pcs:ClientParticipant xmlns:pcs="...">
      <wsa:EndpointReference>...</wsa:EndpointReference>
      <pcs:ProcessReference pcs:ProcessNS="..." pcs:ProcessName="..." />
    </pcs:ClientParticipant>
  </wsp:AppliesTo>
  <wsp:Policy>...</wsp:Policy>
</wsp:PolicyAttachment>
```

Figura 3. Exemplo de uma anexação de uma política ao sujeito participante cliente

De maneira geral, a etapa de composição das políticas consiste nos seguintes passos, descritos também na Figura 4:

1. analisar a descrição do processo de negócio a fim de identificar todas as trocas de mensagem entre dois participantes;
2. para cada troca de mensagem do processo:
   (a) identificar qual a mensagem trocada;
   (b) identificar o cliente e o provedor da operação usada na troca de mensagem;
   (c) Identificar e confrontar as políticas:
      i. se um dos participantes da troca de mensagem for descoberto dinamicamente, a troca de mensagem é assinalada como **dinâmica**;
      ii. em caso contrário:

A. calcular a intersecção da política de qualidade de proteção aplicada pelo provedor à mensagem com a política aplicada pelo cliente à sua participação no processo;
B. anexar a política resultante da intersecção à troca de mensagem;

3. calcular o resultado sumário:
   (a) se alguma política resultante for nula (não contiver nenhuma alternativa), o resultado sumário da composição é de possibilidade de falha na execução do processo;
   (b) se não houver nenhuma política nula:
      i. se pelo menos uma troca de mensagem for dinâmica, o resultado sumário é inconclusivo;
      ii. em caso contrário o resultado sumário é sucesso.

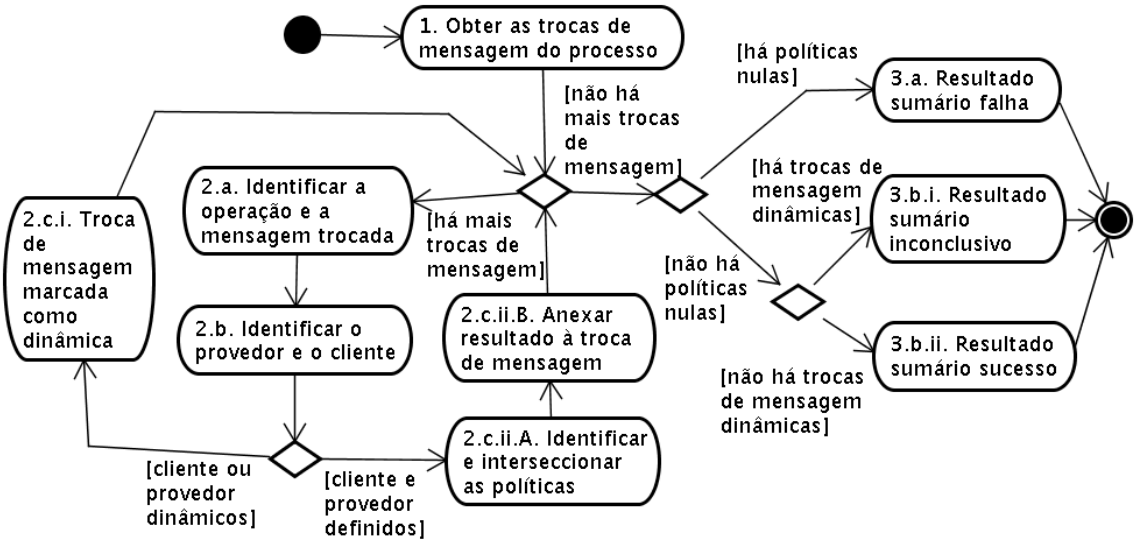

Figura 4. Atividades da etapa de composição de políticas

Os **passos 1, 2.a e 2.b** são dependentes do tipo de processo e precisam ser diferenciados para WS-BPEL e WS-CDL. A WS-BPEL apresenta diversas formas de interação, cada uma com características de fluxo de controle diferentes. As atividades *invoke*, *receive* e *reply* são seqüenciais e síncronas; a atividade *pick* permite a definição de diversos tratadores de mensagens ativos simultâneamente e termina quando um desses tratadores dispara; os tratadores de eventos simplesmente executam uma atividade quando uma mensagem é recebida, paralelamente à execução das atividades do processo. A operação invocada, em todos esses casos, é identificada pelos atributos `partnerLink`, `portType` e `operation` dos elementos XML que as representam. Na atividade *invoke*, o serviço orquestrador é o cliente e o *partnerLink* indica o serviço provedor. Em todas as outras interações, o serviço orquestrador é o provedor da operação e o *partnerLink* indica o cliente. A mensagem trocada, em todos esses casos, pode ser deduzida a partir da operação e do tipo de atividade.

Já na WS-CDL, todas as interações são descritas por uma mesma construção, o elemento `<interaction>` que descreve uma seqüência de envios de mensagens e cada envio é descrito por um elemento `<exchange>`. O serviço provedor da operação é sempre aquele que disponibiliza o canal em que ocorrem as trocas de mensagens,

indicado pelo atributo `channelVariable` do elemento `<interaction>` e pelo atributo `toRoleTypeRef` do elemento `<participate>`. O cliente é indicado pelo atributo `fromRoleTypeRef` do mesmo elemento. A operação é indicada pelo atributo `operation` do elemento `<interaction>` e, em cada elemento `<exchange>`, a mensagem é identificada pelo atributo `action`.

O **passo 2.c** é o núcleo do modelo de composição de políticas e consiste no cálculo da política resultante. A única complicação ocorre quando pelo menos um dos participantes envolvidos na troca de mensagem só será identificado no momento da execução do processo. Nesses casos, o serviço compositor não assume nada com relação à compatibilidade das políticas e apenas assinala a troca de mensagem como "dinâmica". O conjunto das trocas de mensagem dinâmicas é adicionado à resposta gerada pelo serviço compositor juntamente com a política composta. Quando todos os participantes são conhecidos previamente, o serviço compositor obtém as políticas aplicadas à mensagem na WSDL do serviço que provê a operação e as políticas aplicáveis à participação do cliente no processo de negócio, depois aplica a operação de intersecção de políticas definida na *WS-Policy*. O resultado da intersecção conterá a coleção das alternativas de política compatíveis dos dois operandos. Se não houver compatibilidade, essa coleção será vazia e a política resultante será nula.

Para anexar as políticas às trocas de mensagens (**passo 2.c.ii.B**), o serviço compositor usa o elemento `<wsp:PolicyAttachment>` da especificação *WS-PolicyAttachment* [WS-POLICYATTACHMENT 2007]. Essa construção, como exemplificado na Figura 5, deve conter o alvo da anexação dentro do elemento `<wsp:AppliesTo>`, seguido da política e dos mecanismos de segurança aplicados ao anexo. Para o presente modelo, o conteúdo do elemento `<wsp:AppliesTo>` deve conter a identificação da troca de mensagem a qual se aplica a política. Uma forma genérica de identificar a interação é por meio de expressões *XPath* [XPATH 1999] que apontem para os elementos que representam as interações, como está exemplificado na Figura 5. A parte (a) da figura mostra um anexo a um *receive* WS-BPEL. A parte (b) é semelhante e se aplica a um elemento `<exchange>` contido em uma coreografia WS-CDL.

Além das políticas resultantes das intersecções e das trocas de mensagem dinâmicas, o serviço compositor gera também um resultado sumário, que indica se a execução do processo tem ou não risco de falhar por incompatibilidade dos requisitos de segurança. Um **resultado sumário de sucesso** indica que todos os participantes do processo concordam com relação aos mecanismos de proteção usados nas trocas de mensagens. Um **resultado sumário de falha** indica que há um ou mais pontos da colaboração nos quais os participantes não concordam quanto aos mecanismos de proteção das mensagens, entretanto, não indica que o processo nunca poderá ser executado com sucesso, pois uma dada interação, descrita no processo, pode não ser executada graças à desvios no fluxo da execução (condicionais, exceções, etc.). Ainda, um **resultado sumário inconclusivo** indica que nenhuma política nula foi encontrada, mas há pelo menos uma troca de mensagem na qual ao menos um dos participantes é descoberto dinamicamente, o que indica que o serviço compositor não pode garantir a compatibilidade dos requisitos de segurança.

4. Implementação

Um protótipo envolvendo o modelo proposto neste trabalho foi definido e implementado visando comprovar a sua flexibilidade, bem como a viabilidade em processos de negó-

```
<!-- (a) -->
<wsp:PolicyAttachment xmlns:wsp="...">
  <wsp:AppliesTo>
    <pcs:ExchangeSubject>
      <pcs:ProcessReference pcs:ProcessNS="..." pcs:ProcessName="..." />
      <pcs-bpel:Receive xmlns:pcs-bpel="..." pcs-bpel:Path="..." />
    </pcs:ExchangeSubject>
  </wsp:AppliesTo>
  <wsp:Policy>...</wsp:Policy>
</wsp:PolicyAttachment>
<!-- (b) -->
<wsp:PolicyAttachment xmlns:wsp="...">
  <wsp:AppliesTo>
    <pcs:ExchangeSubject>
      <pcs:ProcessReference pcs:ProcessNS="..." pcs:ProcessName="..." />
      <pcs-cdl:Exchange xmlns:pcs-cdl="..." pcs-cdl:Path="..." />
    </pcs:ExchangeSubject>
  </wsp:AppliesTo>
  <wsp:Policy>...</wsp:Policy>
</wsp:PolicyAttachment>
```

Figura 5. Exemplos de Anexação de Política em Processos WS-BPEL e WS-CDL

cios baseados em composição de Serviços *Web*. Até o presente momento, o protótipo implementa apenas a composição de políticas em processos WS-BPEL.

Para compor a camada necessária para o desenvolvimento de aplicações baseadas na arquitetura dos Serviços *Web*, escolheu-se o *framework* de código aberto *Apache Axis2* e o *Apache Tomcat*[4] como servidor de aplicação. Para compor a camada de qualidade de proteção e prover segurança as mensagens trocadas entre os participantes da composição de serviços, adotou-se o módulo para o padrão *WS-Security* do Axis2 *Apache Rampart*[5]. Este módulo, baseado na biblioteca *Apache WSS4J*, oferece ainda suporte a aplicação de políticas descritas segundo a *WS-SecurityPolicy*. Para a camada de descrição de processos de negócios, adotou-se a ferramenta *Apache Orchestration Director Engine*[6] (Apache ODE), que implementa a WS-BPEL. No momento, apenas a versão 1.1 da WSDL é suportada no protótipo, por meio da biblioteca WSDL4J[7].

Ainda no escopo da camada de qualidade de proteção, inicialmente, havia sido estipulado que a biblioteca *Apache Neethi*[8] seria usada para processar políticas *WS-Policy* no protótipo desenvolvido. No decorrer do desenvolvimento, no entanto, observou-se que tal biblioteca não se adequava às necessidades do modelo, já que esta não implementa a intersecção de políticas e não possibilita interpretar os anexos de política em WSDL, funções necessárias para o modelo proposto. Por esse motivo, decidiu-se implementar uma biblioteca que contemplasse as especificações *WS-Policy*, *WS-PolicyAttachment* e *WS-SecurityPolicy* e que oferecesse as funcionalidades necessárias para verificação de compatibilidade. Flexibilidade e extensibilidade foram os objetivos principais na concepção e elaboração desta biblioteca, denominada de *GWSPolicy* (*GCSeg WS-Policy Implementation*), sendo que esta pode ser usada independente do protótipo desenvolvido.

[4]http://ws.apache.org/axis2 e http://tomcat.apache.org, respectivamente.
[5]http://ws.apache.org/axis2/modules/rampart/1_2/security-module.html
[6]http://ode.apache.org/
[7]http://sourceforge.net/projects/wsdl4j
[8]http://ws.apache.org/commons/neethi/

O serviço compositor de políticas, que também se encontra na camada de qualidade de proteção, implementa apenas uma operação que recebe como entrada uma mensagem contendo a descrição do processo, uma lista de participantes predefinidos e uma lista de parâmetros adicionais. A resposta gerada pelo serviço compositor contém um atributo indicando o resultado sumário, uma lista de referências para as trocas de mensagens dinâmicas e uma lista de anexos de política que formam a política composta do processo.

O serviço compositor de políticas foi implementado de forma a ser extensível para diversos formatos de descrição de processos de negócio. Por esse motivo, a lógica do protótipo foi dividida entre uma parte genérica, independente da linguagem de descrição do processo de negócio e uma parte específica, contendo módulos para cada linguagem de descrição de processo implementada.

A parte independente da linguagem de descrição é responsável por obter e analisar as descrições WSDL e as políticas *WS-Policy* dos participantes, calcular a intersecção das políticas, além de coordenar as etapas da composição. Cada módulo específico de uma linguagem de descrição é responsável por analisar uma descrição de processo e obter a lista de trocas de mensagens declaradas na descrição, bem como gerar as referências para as partes do processo que representam as trocas de mensagens. Um módulo para analisar processos WS-BPEL foi implementado, tendo como base a biblioteca *Apache ODE*. Essa biblioteca implementa a análise sintática de processos, bem como um motor de orquestração, que não foi usado no protótipo.

5. Trabalhos relacionados

O trabalho [CHARFI e MEZINI 2005] trata da implementação de aspectos de segurança em uma composição usando WS-BPEL. No entanto, a preocupação maior é com a aplicação das restrições de segurança por parte do motor de orquestração. A especificação de requisitos de segurança por meio de políticas e a verificação de compatibilidade entre políticas de participantes distintos são abordadas apenas superficialmente e apontadas como trabalhos futuros, enquanto, no modelo definido neste artigo, esta é a questão principal. Além disso, diferentemente do modelo proposto neste trabalho, em [CHARFI e MEZINI 2005] apenas orquestrações são contempladas.

Em [HUANG 2005], um *framework* de segurança baseado em políticas, que usa descrições semânticas de requisitos de segurança, é modelado. O *framework* prevê a especificação de políticas de alto nível, incluindo restrições complexas relacionadas com a lógica de negócio, por meio de descrições semânticas. O modelo está focado na WS-BPEL e prevê a integração de uma definição de processo com as políticas de segurança, no entanto, não define como isso deve ser feito, nem como as políticas devem ser especificadas. O modelo aqui proposto, por outro lado, não está restrito à WS-BPEL e está baseado no uso do padrão *WS-Policy* para expressar e comparar requisitos de segurança.

Em [CARMINATI et al. 2006], é elaborado um modelo de composição de Serviços *Web* com base em aspectos de segurança. A abordagem é baseada na descrição de requisitos e propriedades de segurança por meio de políticas, sendo que a avaliação da compatibilidade entre os requisitos é executada por um serviço chamado *Security Matchmaker*. No entanto, ao contrário do modelo proposto neste trabalho, esse modelo não prevê o uso da *WS-Policy* para descrever as políticas. Além disso, [CARMINATI et al. 2006] propõe uma arquitetura de serviços para escolher os participantes da composição com base em requisitos de segurança. Novamente, esse trabalho está focado somente em orquestra-

ções WS-BPEL e, ao contrário do modelo proposto, não trata de coreografias WS-CDL.

Alguns trabalhos [CHARFI et al. 2006, SONG et al. 2006] tratam da implementação de mecanismos de segurança em processos de negócio de forma dinâmica, sem tratar, no entanto, da verificação de compatibilidade de requisistos dos serviços envolvidos. Outros trabalhos [ZHU et al. 2006, SRIVATSA et al. 2007, ROUACHED e GODART 2007] tratam da especificação de políticas de controle de acesso e autorização para composições de Serviços *Web*, mas não tratam das políticas de qualidade de proteção.

6. Conclusão

A composição de Serviços *Web* já está especificada em padrões bem aceitos ou propostas promissoras (WS-BPEL e WS-CDL) e também existem diversas especificações que definem como as propriedades de segurança devem ser atendidas nos Serviços *Web* simples. No entanto, não há especificação que indique como tratar as políticas de qualidade de proteção dos participantes de uma composição de Serviços Web.

Sem uma verificação anterior à execução do processo, cada Serviço *Web* fará a aplicação da sua política de forma independente, no momento da execução, o que não é adequado, a menos que os parceiros sejam descobertos durante a execução do processo. Toda a lógica de negócio pode ser colocada em risco se algum requisito de segurança não puder ser cumprido. Há casos ainda, como na formação de uma organização virtual, em que os parceiros de negócio são selecionados tendo como base seus requisitos e mecanismos de segurança. Nesses casos, se faz necessário um meio de identificar, antes da execução do processo, em quais pontos não há compatibilidade entre as políticas de qualidade de proteção dos participantes da colaboração.

Este artigo descreveu um modelo, baseado nos padrões de Serviços *Web* e de processos de negócio, para a composição de políticas de qualidade de proteção dos serviços envolvidos no processo de negócio. Tal modelo permite que as partes envolvidas no processo saibam se suas restrições de segurança são compatíveis com as restrições dos outros parceiros. Esse conhecimento facilita o desenvolvimento de processos de negócios em sistemas abertos, pois automatiza uma verificação necessária para a construção de uma colaboração segura.

Quando comparado aos trabalhos relacionados, pode-se destacar como contribuição do modelo proposto o fato deste oferecer um serviço para verificação de compatibilidade de políticas de qualidade de proteção antes da execução dos processos de negócio, inclusive com suporte a verificação mesmo diante de composições dinâmicas. Outros dois pontos fortes do modelo são: o uso de um padrão consolidado para especificação de políticas (WS-Policy) e o suporte não somente a linguagem de orquestração WS-BPEL, mas também a linguagem WS-CDL para descrição dos processo de negócios. O protótipo do modelo foi construído com base em bibliotecas e ferramentas abertas e tanto a biblioteca GWSPolicy quanto o Serviço Compositor de Políticas foram desenvolvidos para serem flexíveis, extensíveis e estarem em conformidade com os padrões de Serviços *Web*.

Como trabalhos futuros, tem-se como meta finalizar a implementação do protótipo, para que este suporte coreografias WS-CDL e descrições WSDL 2.0, além de realizar avaliações de desempenho e segurança. Também se propõe pesquisar e desenvolver mecanismos para tratar as incompatibilidades de políticas de qualidade de proteção encontradas em um processo de negócio através de técnicas de transposição de tecnologias

de segurança e de técnicas baseadas em uma terceira-parte confiável.

Referências

CARMINATI, B., FERRARI, E., e HUNG, P. C. K. (2005). Web service composition: A security perspective. In *Proceedings WIRI*, pages 248 –253.

CARMINATI, B., FERRARI, E., e HUNG, P. C. K. (2006). Security conscious web service composition. In *ICWS'06: Proceedings of the IEEE International Conference on Web Services*. IEEE.

CHARFI, A. e MEZINI, M. (2005). Using aspects for security engineering of web service compositions. In *Proceedings of the 2005 IEEE International Conference on Web Services*, volume I, pages 59–66. IEEE.

CHARFI, A., SCHEMLING, B., HEINZREDER, A., e MEZINI, M. (2006). Reliable, secure, and transacted web service compositions with ao4bpel. In *Proceedings of the European Conference on Web Services (ECOWS'06)*. IEEE.

HUANG, D. (2005). Semantic policy-based security framework for business processes. In *4th International Semantic Web Conference*.

PELTZ, C. (2003). Web services orchestration and choreography. *IEEE Computer*, 36(10):46–52.

ROUACHED, M. e GODART, C. (2007). Specification and verification of authorization policies for web services composition. In *CAISE Forum, CEUR Workshop Proc.*

SONG, H., Sun, Y., YIN, Y., e ZHENG, S. (2006). Dynamic weaving of security aspects in service composition. In *Proceedings of the Second IEEE International Symposium on Service-Oriented System Engineering (SOSE'06)*. IEEE.

SRIVATSA, M., IYENGAR, A., MIKALSEN, T., ROUVELLOU, T., e YIN, J. (2007). An access control system for web service compositions. In *Web Services, 2007. ICWS 2007. IEEE International Conference on*.

WANGHAM, M. S., de MELLO, E. R., RABELLO, R., e FRAGA, J. S. (2005). Provendo garantias de segurança para formação de organizações virtuais. *Gestão Avançada de Manufatura*, 22:75–84.

WS-ADDRESSING (2006). Web services addressing 1.0 - core. W3C Recommendation. http://www.w3.org/TR/2006/REC-ws-addr-core-20060509.

WS-ARCHITECTURE (2004). Web services architecture. W3C Working Group Note. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/.

WS-BPEL (2007). Web services business process execution language version 2.0. OASIS Standard. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

WS-CDL (2005). Web services choreography description language version 1.0. W3C Candidate Recommendation. http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/.

WS-FEDERATION (2006). Web services federation language (ws-federation) version 1.1. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-Federation-V1-1B.pdf.

WS-METADATAEXCHANGE (2006). Web services metadata exchange (ws-metadataexchange) version 1.1. http://www.w3.org/TR/2006/REC-ws-addr-core-20060509.

WS-POLICY (2007). Web services policy 1.5 - framework. W3C Recommendation. http://www.w3.org/TR/2007/REC-ws-policy-20070904/.

WS-POLICYATTACHMENT (2007). Web services policy 1.5 - attachment. W3C Recommendation. http://www.w3.org/TR/2007/REC-ws-policy-attach-20070904/.

WS-SECURECONVERSATION (2007). Ws-secureconversation 1.3. OASIS Standard. http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.pdf.

WS-SECURITY (2006). Web services security: Soap message security 1.1. OASIS Standard Specification. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf.

WS-SECURITYPOLICY (2007). Ws-securitypolicy 1.2. OASIS Standard. http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.pdf.

WS-TRUST (2007). Ws-trust 1.3. OASIS Standard. http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf.

XPATH (1999). Xml path language (xpath) version 1.0. W3C Recommendation. http://www.w3.org/TR/1999/REC-xpath-19991116.

ZHU, J., ZHOU, Y., e TONG, W. (2006). Access control on the composition of web services. In *NWESP '06: Proc. of the International Conference on Next Generation Web Services Practices*, pages 89–93, Washington, DC, USA. IEEE Computer Society.