Uma abordagem para inclusão de segurança na descrição semântica de serviços Web no contexto do SISCEAB

Gerson Monteiro Siqueira[1], Alexandre de Barros Barreto[2], José Parente de Oliveira[1] e Inaldo Costa[13]
1. Instituto Tecnológico de Aeronáutica (ITA) – Avenida Marechal Eduardo Gomes, S/N° – DCTA – São José dos Campos – SP.
2. Instituto de Controle do Espaço Aéreo (ICEA) - Praça Marechal-do-Ar Eduardo Gomes, 50, São José dos Campos – SP.
3. Universidade Federal do Maranhão (UFMA) - Av. dos Portugueses, 1966 - Bacanga, São Luís – MA.

***Resumo* — Cada vez mais aumenta a demanda de usuários para acessar serviços e informações na Web de forma mais padronizada, automatizada e segura. Neste contexto, estão também os serviços de Tráfego Aéreo, a nível mundial. Como solução, a ICAO estabeleceu a arquitetura SWIM, baseada em SOA, cuja a tecnologia mais utilizada é o serviço Web, como o novo modelo de padrões, infraestrutura e governança no ambiente de Gerenciamento de Tráfego Aéreo (ATM). Porém, os serviços Web apresentam alguns problemas de interação, sendo necessário utilizar a semântica para melhorar a sua segurança e interoperabilidade. Objetivando contribuir com o cenário brasileiro, a finalidade deste artigo é apresentar uma abordagem para a inclusão de segurança na descrição semântica de serviços Web, dentro da conjuntura do SISCEAB, demonstrando a integração de ontologias de segurança, já existentes, com ontologias de serviços descritas em OWL-S.**

***Palavras-Chave* — OWL-S, SWIM, SISCEAB.**

I. INTRODUÇÃO

Atualmente, as organizações buscam disponibilizar na Web informações e serviços, de forma mais padronizada, automatizada e segura possível, usando para isso ambientes heterogêneos de Tecnologia da Informação (TI). Como consequência, necessitam garantir a interoperabilidade e a segurança desses sistemas.

Nesse contexto, o conceito SOA (*Service-Oriented Architecture*), que é um modelo de referência da OASIS (*Advancing Open Standards for the Information Society*) [1], permite estruturar, num ambiente de computação distribuída, aplicações de *software* em elementos chamados serviços [2], de forma a apoiar os processos de negócio das organizações.

A tecnologia de serviços Web é a mais usada na construção de soluções SOA [2]. Porém, os padrões definidos para os serviços Web não tratam de forma adequada os problemas de interoperabilidade, uma vez que garantem a automatização dos serviços usando apenas informações sintáticas, o que torna as buscas de serviços pouco eficazes [3]. Para resolver tais problemas, tem-se incorporado aos serviços anotações semânticas e o uso de ontologias para descrever conceitos, o que é conhecido como Web Semântica [4]. Este fato possibilita que os serviços possam ser descritos de forma mais abrangente, sejam em seus aspectos funcionais e não funcionais, como, por exemplo, as suas características de segurança [5-6], assim como o emprego de mecanismos eficientes para descoberta, seleção, composição, invocação e monitoramento [5]. Uma das tecnologias mais utilizadas para a descrição semântica de serviços Web é a linguagem OWL-S (*Web Ontology Language for Service*) [3-7].

Em virtude da necessidade de disponibilização de informações em ambientes e atores heterogêneos, a Organização de Aviação Civil (OACI) estabeleceu dentro do Doc. 9750 - *Global Air Navigation Plan* – 2013-2028 (4th Edition-2013) [8], entre suas prioridades para a construção de um ambiente mais seguro para a aviação mundial, que o novo modelo de padrões, infraestrutura e governança para o compartilhamento de informações aeronáuticas seria implantado usando o paradigma baseado em serviços, o que é conhecido como a arquitetura SWIM (*System Wide Information Management*) [8]. No Brasil, o Departamento de Controle do Espaço Aéreo (DECEA), órgão central do Sistema de Controle do Espaço Aéreo Brasileiro (SISCEAB) [9], é o responsável por acompanhar a implantação mundial do SWIM, bem como estabelecer diretrizes visando a total aderência ao estabelecido nas recomendações da OACI [10].

Através do supracitado ambiente, os diversos usuários de informações aeronáuticas (aeronaves, órgãos de controle, etc.) buscarão estruturar suas decisões através do consumo de informações disponibilizadas pelos prestadores de serviços existentes (companhias aéreas, prestadores de serviço de navegação aérea, serviços de meteorologia, etc.) [11].

Em virtude da heterogeneidade dos prestadores de serviço, no que diz respeito a natureza do serviço, nível de confiabilidade e padronização da informação, é necessário que exista uma representação de conhecimento comum, permitindo que o entendimento semântico da informação seja unificado. Adicionalmente, uma vez que as supracitadas informações irão suportar processos decisórios críticos, como por exemplo, a decisão de realizar ou não um pouso, é importante garantir as suas integridades e autenticidades [12].

Nessa conjuntura, expandido o estudo realizado em [13], o presente artigo apresentará uma abordagem de inclusão de segurança na descrição semântica de serviços Web, através da linguagem OWL-S, dentro do contexto do SISCEAB, demonstrando a viabilidade e a importância da integração de ontologias de segurança com ontologias de serviços, que poderá permitir a definição de requistos e capacidades de segurança dos mesmos. Este trabalho visa, ainda, contribuir para os estudos e as pesquisas realizadas pelo DECEA, em seu programa de implantação da arquitetura SWIM, no que se refere ao processo de descoberta e seleção de serviços Web.

As demais partes deste artigo estão organizadas da seguinte forma: Seção 2 - Conceituação Preliminar, onde são

detalhados os principais conceitos. Na Seção 3 é apresentada uma revisão de literatura, dos estudos de casos relacionados. Na Seção 4 é apresentada a descrição semântica de serviços Web. A Seção 5 traz os experimentos e análises dos resultados de um estudo de caso específico utilizando-se um serviço Web de Meteorologia. Finalmente, algumas considerações finais são apresentadas.

II. Conceituação Preliminar

Serviço Web

Os serviços Web são sistemas de *software* baseados na estrutura SOA, projetados para permitir a interação e a interoperabilidade entre as diferentes aplicações Web, independentemente das suas plataformas de desenvolvimento e sistemas operacionais suportados, através de um ambiente computacional distribuído, onde eles são identificados através de uma URI (*Uniform Resource Identifier*) [14].

Para suportar seu funcionamento, os serviços Web usam os seguintes padrões [3]: a) XML (*Extensible Markup Language*): que é uma linguagem de marcação utilizada na troca de informações; b) SOAP (*Simple Object Access Protocol*): que consiste em um protocolo de comunicação baseado em XML, utilizado para realizar a troca de mensagens; c) WSDL (*Web Services Definition Language*): que é uma linguagem baseada em XML que descreve o serviço, definindo sua estrutura e conteúdo, bem como as operações suportadas, sua localização e as formas de transmissão das mensagens; e d) UDDI (*Universal Discovery Description Integration*): que é um mecanismo que possibilita a publicação e a busca dos serviços, visando sua posterior utilização.

A Fig. 1 representa o ciclo de vida de um serviço *Web*, baseada na interação de três agentes de *software*: Provedor de Serviços, Consumidor de Serviços e Registro de Serviços, através de quatro operações: Descrição (o provedor descreve o serviço em WSDL), Publicação (o provedor faz o registro do serviço através do UDDI), Descoberta (o consumidor realiza a busca do serviço e, após a sua descoberta, ou seleção, recebe o respectivo WSDL) e Invocação (utilizando o WSDL recebido, o consumidor invoca a execução do serviço ao provedor, através do SOAP) [3]:

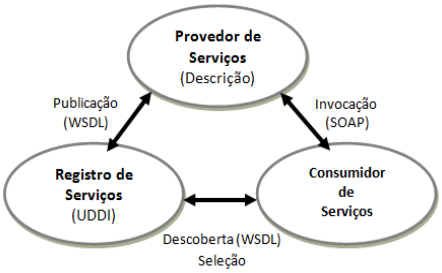

Fig. 1. Arquitetura Básica do Ciclo de Vida de um serviço Web [3].

Serviço Web Semântico

A Web Semântica pode ser definida como uma evolução do *World Wide Web (WWW)*, onde a informação disponível possui semânticas acessíveis e executadas por máquinas para aumentar o processamento automatizado de informações e a interoperabilidade dos sistemas de informação [15], superando a limitação da abordagem sintática da Web atual [16].

No contexto da Web Semântica, o conceito de ontologia é bastante empregado, cujo principal objetivo é a organização de um vocabulário representativo de uma área de conhecimento e o seu compartilhamento por meio do uso de conceitos de taxonomia, relações e regras (axiomas) referentes a um domínio considerado [3]. É também definida como uma especificação formal e explícita de uma conceitualização compartilhada [17].

As ontologias são modeladas por meio de linguagens de marcação semântica, baseadas em lógica computacional. A OWL (*Web Ontology Language*) [18], recomendada pelo W3C (*World Wide Web Consortium*), é uma das linguagens mais utilizadas, tendo em vista o seu grande poder de expressividade [3].

OWL-S

Uma das aplicações da OWL resultou na especificação de uma sublinguagem para descrição de serviços Web semânticos, chamada OWL-S [19], que é um conjunto de ontologias que descreve serviços Web, sendo, atualmente, uma das tecnologias mais consolidadas [7]. A OWL-S é definida através de uma ontologia principal, *Service*, e outras três sub-ontologias: *ServiceProfile*, *ServiceModel* e *ServiceGrounding*, respectivamente [16].

A função de *Service* é interligar as três sub-ontologias, através da definição de classes abstratas que serão especializadas por cada ontologia interligada [7]. A classe abstrata *Service* é definida pela ontologia *Service*, e fornece um ponto organizacional de referência para o serviço Web. Uma instância de *Service* existirá para cada serviço distinto publicado. Cada instância de *Service* apresentará uma descrição *ServiceProfile*, será detalhada por uma descrição *ServiceModel*, e apoiará uma descrição *ServiceGrounding* [3-19]. A Fig. 2 apresenta a estrutura de *Service*, com as suas propriedades *presents*, *describedby* e *supports*, com os seus respectivos ranges:

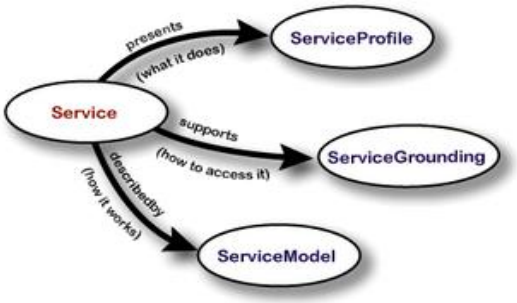

Fig. 2. Estrutura da Ontologia *Service* [19].

A especialização concreta de *ServiceProfile* é realizada pela classe *Profile* [7], que representa a descrição geral do serviço Web: o que ele realiza, as suas restrições, qualidade de serviço, requisitos e outras características funcionais e não funcionais [19]. O *Profile* é utilizado no processo de publicação e a descoberta do serviço web. A Fig. 3 apresenta a estrutura de *Profile*, com as suas subclasses e propriedades.

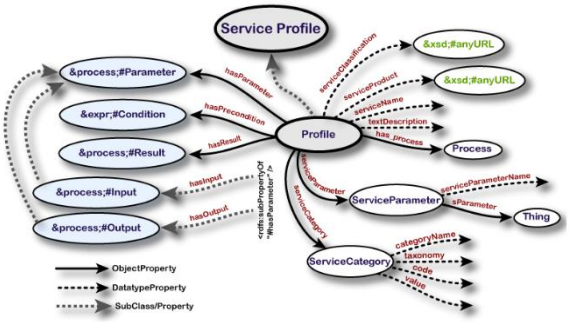

Fig. 3. Estrutura da Ontologia *ServiceProfile* [19].

A especialização concreta de *ServiceModel* é realizada pela classe *Process* [7], que descreve detalhes de como um serviço será utilizado [19]. A classe *Process* se relaciona com os parâmetros do serviço Web chamados de IOPE: Entrada (*Input*), Saída (*Output*), Pré-condições (*Preconditions*) e Efeitos (*Effects*). Com base nas descrições semânticas dos IOPE é possível garantir a interoperabilidade entre os processos de serviços Web [7].

A ontologia *ServiceModel* modela os serviços como processos, sendo definidos em três tipos, que são subclasses da classe *Process* [3]: a) **Atômico**: são executados em uma única interação, não tem subprocessos e são invocados diretamente; b) **Simples**: é parecido com um processo atômico, porém não é invocável e é utilizado somente como elemento de abstração; c) **Composto**: podem ser decompostos em outros processos, que podem ser compostos ou não. É possível ter um número ilimitado de processos compostos aninhados.

A especialização concreta de *ServiceGrounding* é realizada pela classe *Grounding* [7], que especifica os tipos de protocolos de comunicação, formatos de mensagens, e outros detalhes específicos do serviço (Ex. portas utilizadas), para permitir que um agente possa interagir e invocar um determinado serviço [19]. O objetivo principal do *Grounding* é mapear as informações do WSDL, como , por exemplo, as informações abstratas de entrada e saída de um processo atômico, para que as mesmas possam ser representadas concretamente na forma de mensagens [3-19].

Segurança de serviços web

As maiores preocupações no uso da tecnologia de serviços Web diz respeito em como prover um serviço de forma segura [20]. Isso é desenvolvido através do uso de atributos não funcionais. O *WS-Security* [21] e *WS-SecurityPolicy* [22] são os padrões mais aceitos para a definição e o processamento da segurança de mensagens em serviços Web. O *WS-Security* define a sintaxe dos elementos de segurança na mensagem SOAP, e o *WS-SecurityPolicy* especifica quais medidas de segurança serão aplicadas nas partes da mensagem (cabeçalho e corpo) [23]. O *WS-SecurityPolicy* é construído em cima do framework *WS-Policy* [21-23], que é uma estrutura de uso geral que permite especificar, em XML, políticas de serviços, relacionadas com atributos não-funcionais, como por exemplo, segurança [24]. Entretanto, o *WS-Policy* representa sintaticamente os aspectos das propriedades e atributos, não possuindo semântica, o que o torna ineficiente na verificação de compatibilidades entre as políticas de cliente e provedores de serviços, no processo de descoberta e seleção dos mesmos, pois pode gerar resultados não confiáveis [25]. Nesse aspecto, estudos vem sendo realizados para a utilização de ontologias na descrição da segurança dos serviços Web [5-6-20-25].

Arquitetura SWIM

O SWIM é um *framework* definido pela OACI, composto por padrões, infraestrutura e governança, baseado em serviços, que permite compartilhar, em tempo real, informações de forma descentralizada e rápida, viabilizando maior facilidade e eficiência na execução das operações relacionadas com o Gerenciamento de Tráfego Aéreo. O seu modelo de referência de informações ATM (*ATM Information Reference Model* - AIRM) possibilita o compartilhamento e a troca de informações através da harmonização global de padrões, por meio de modelos conceituais e lógicos definidos para os diversos tipos de informações aeronáuticas, como por exemplo: Modelo de Troca de Informação Aeronáutica (*Aeronautical Information eXchange Model* - AIXM), Modelo de Troca de Informação de Voo (*Flight Information eXchange Model* - FIXM) e Modelo de Troca de Informação de Clima/Tempo (*Weather Information eXchange Model* - WIXM). Para a implantação do SWIM, outros elementos foram definidos, como o Modelo de Referência de Serviço de Informação (*Information Service Reference Model* - ISRM) que estabelece a repartição lógica do serviço de informação e os seus padrões de comportamento, como por exemplo a Qualidade de Serviço (QoS), e as Funções de Gerenciamento de Informação, que inclui a governança, como a gestão sobre a identificação do usuário, descoberta de recursos, a harmonização das regras de propriedade, fornecimento e utilização de dados, definição e aplicação dos Acordos de Nível de Serviço (SLA) entre os diferentes participantes, aspectos de segurança, como autenticação, criptografia e autorização, dentre outras [26]. A Fig. 4 representa a ideia de compartilhamento descentralizado de informações, entre os potências usuários do SWIM.

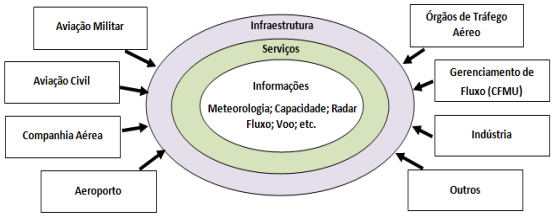

Fig. 4. Arquitetura SWIM.

III. Revisão da Literatura

Vários trabalhos tratam do uso de ontologias e regras semânticas para fortalecer a segurança dos serviços Web.

Em [6], foram apresentadas ontologias modeladas em *DAM-L + OIL* (antecessora da *OWL-S*), que descrevem, num alto nível de abstração, alguns mecanismos e recursos de segurança mais comumente utilizados, que são referenciados, no *Profile* dos serviços, através de propriedades que representam os seus requisitos e capacidades de segurança.

Tais propriedades viabilizam a descoberta e a seleção dos serviços, por meio de um *reasoner* de segurança, composto pelo sistema de raciocínio *JTP* (*Java Theorem Prover*) e um algoritmo de correspondência (*matching)*, que considera, além das descrições funcionais do serviço Web desejado, o grau de correspondência entre as características não funcionais de segurança dos provedores e agentes solicitantes de serviços.

Ainda nesse trabalho, foi realizada a interligação entre as ontologias de segurança e o linguagem DAM-L, através da ontologia *serviceSecurity.owl*, que define a classe *SecurityMechanism*, raiz da ontologia *security.owl*, como uma subclasse da classe *ServiceParameter*, que possui um relacionamento com classe *Profile* por meio da propriedade *serviceParameter*. Os requisitos e as capacidades de segurança são representados semanticamente por duas propriedades objeto criadas para os serviços Web, *securityRequirement* e *securityCapability*, que são subpropriedades de *serviceParameter*.

Em [20], é desenvolvida uma extensão do seu estudo realizado em [6], onde é abordada a formalização de políticas de privacidade e autorização para invocação de serviços Web, através do fornecimento de ontologias para anotação de parâmetros de entrada e saída de serviços Web, considerando recursos de segurança de criptografia e assinatura digital, e da utilização da linguagem Rei [27], baseada em RDF *Schema*, utilizada para especificação e formalização de políticas, e definição de regras e restrições sobre ontologias de domínio. Para a utilização de serviços Web, considerando as políticas implantadas, foi apresentada uma ferramenta composta pelo OWL-S *Matchmaker*, que considera as características e os parâmetros funcionais e de segurança, descritos no *profile* do serviço, para a busca e seleção do mesmo, e por uma máquina virtual proposta, denominada OWL-S (VM), que é um processador de OWL-S, que se baseia no *process model* e *grounding*, para realizar a invocação de serviços Web.

Em [5], é apresentada a descrição semântica de serviços básicos de segurança, como autenticação, criptografia e assinatura digital, especificados em OWL-S, que permite a reutilização dos mesmos na composição de serviços mais complexos, visando atender as necessidades das políticas de segurança dos clientes ou provedores de serviços Web, respectivamente. Como exemplo, foi demonstrada parte de um serviço semântico de assinatura XML.

Em [16], é utilizada a OWL para enriquecer as políticas de segurança e a interoperabilidade dos serviços Web, através da definição de uma ontologia de segurança, num alto nível de abstração, que estabelece conceitos e relacionamentos para a proteção de mensagens trocadas na interação com os referidos serviços, considerando os padrões de assinatura e criptografia XML, do *WS-Security*.

Em [23], é abordada a conversão do padrão sintático de *WS-SecurityPolicy* para um modelo semântico, por meio de uma ontologia em OWL-DL, que é estendida através de regras semânticas adicionais, baseada em *Semantic Web Rule Language* (SWRL), que estabelecem novos relacionamentos na ontologia e permitem especificar e verificar as correspondências semânticas, definidas em quatro possíveis graus de *matching*, entre as políticas de segurança dos provedores e clientes de serviços, de uma forma mais precisa, demonstrando uma melhor garantia de interoperabilidade na utilização de serviços Web.

IV. Descrição Semântica de Serviços Web

Nesta seção, visando contribuir para a futura implantação do SWIM no SISCEAB, no que se refere ao processo de descoberta e seleção de serviços, abordaremos a descrição semântica de serviços Web, a partir de um serviço em WSDL, focando na inclusão de anotações semânticas de parâmetros de segurança na ontologia do serviço descrito.

A Fig. 5 demonstra as fases da metodologia utilizada na abordagem, conforme especificado abaixo:

- Descrição semântica do serviço Web (WSDL), utilizando a linguagem OWL-S;
- Inserção de anotações semânticas de parâmetros de segurança na ontologia do serviço descrito; e
- Teste da ontologia do serviço semântico criado, com as anotações semânticas de segurança.

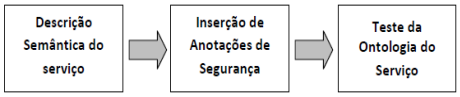

Fig. 5. Fases do processo da descrição semântica.

Para a inserção das anotações semânticas de segurança, serão utilizados os conceitos e propriedades das ontologias de segurança, *"security.owl"* e *"serviceSecurity.owl"*, apresentadas em [6] (Seção III), que definem, respectivamente, os recursos de segurança mais usualmente utilizados, com suas propriedades e instâncias, que podem ser incluídos como anotações semânticas nas ontologias dos serviços Web, e as propriedades *securityCapability* e *securityRequirement*, que permitem referenciar os referidos recursos, no *Profile* dos serviços. A inserção das anotações semânticas de segurança será estabelecida em duas etapas:

Inicialmente, estabelece-se quais mecanismos de segurança serão utilizados na definição das características de segurança do serviço, com base na ontologia *security.owl*. A Fig. 6 representa os mecanismos de segurança da ontologia.

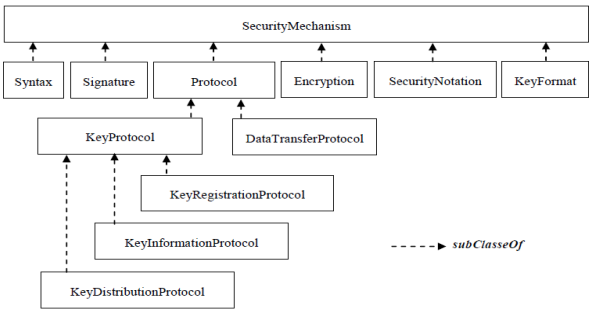

Fig. 6. Ontologia *security.owl* proposta em [6].

Uma vez estabelecidos os mecanismos de segurança utilizados, serão descritos os relacionamentos das propriedades *securityCapability* e *securityRequirement*, cujo o range é a classe *SecurityMechanism*. A Fig. 7 demonstra parte do relacionamento da classe, que será utilizado na nossa descrição semântica.

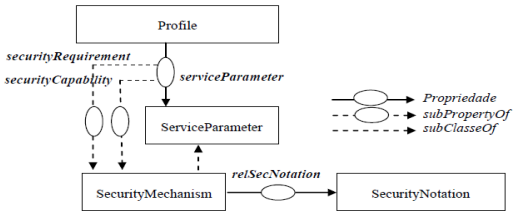

Fig. 7. Parte dos relacionamentos da classe *SecurityMechanism* [6].

Com base nas duas etapas declaradas, anteriormente, estabelecemos as seguintes características de segurança que serão inseridas na ontologia do serviço:

- Requisitos de segurança: o serviço requer que o usuário utilize protocolos de autenticação e autorização para executá-lo.
- Capacidades de segurança: o serviço tem a capacidade de dar suporte de criptografia na comunicação entre o usuário e o provedor.

A Fig. 8 demonstra a descrição textual das anotações semânticas dos recursos de segurança para o estudo de caso, com base na propriedade *relSecNotation*:

```
xmlns:security="http://www.daml.org/services/owl-s/security/security.owl#"
xmlns:serviceSecurity="http://www.daml.org/services/owl-s/security/serviceSecurity.owl#"

<security:KeyProtocol rdf:ID="Sec1">
  <security:relSecNotation rdf:resource=
   "http://www.daml.org/services/owl-s/security/security.owl#Authentication" />
</security:KeyProtocol>

<security:KeyProtocol rdf:ID="Sec2">
  <security:relSecNotation rdf:resource=
   "http://www.daml.org/services/owl-s/security/security.owl#Authorization" />
</security:KeyProtocol>

<security:KeyProtocol rdf:ID="Sec3">
  <security:relSecNotation rdf:resource=
   "http://www.daml.org/services/owl-s/security/security.owl#Encryption" />
</security:KeyProtocol>
```

Fig. 8. Descrição semântica das anotações de segurança.

Para a verificação da ontologia do serviço semântico criado, serão consideradas as seguintes características:

- Funcionais: os parâmetros de Entrada (*Input*) e Saída (*Output*), e as propriedades de tipos de dados "Nome do Serviço" (*serviceName*) e "Descrição do Serviço" (*textDescription*).
- Não funcionais: os parâmetros dos requi*sitos* de segurança e capacidade de segurança.

Para os testes das referidas características serão realizadas consultas à ontologia, através de *queries* SPARQL [28], utilizando a ferramenta *Protégé* [29], aplicação desenvolvida pelas Universidades de *Stanford* e *Manchester*, que permite criar, modelar e verificar ontologias.

V. Experimentação e Análise dos Resultados.

Em [13], foi demonstrada uma arquitetura de serviços Web, modelada em *Service Oriented Architecture Modeling Language* (SOAML), para implantar um ambiente *Airport Collaborative Decision Making (*Airport-CDM*)*, dentro do conceito SWIM, no contexto específico brasileiro. O modelo conceitual foi utilizado para descrever em WSDL um serviço de meteorologia, "*Boletim por Aeródromo*", que tem como parâmetro de entrada a identificação de um aeródromo e como saída o seu boletim meteorológico, conforme Fig. 9.

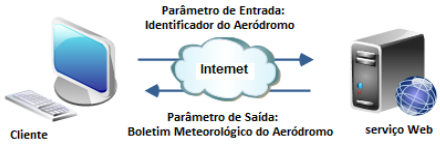

Fig. 9. Execução do serviço Web Boletim por Aeródromo.

Nesta Seção, será apresentado um estudo de caso e os resultados das análises realizadas sobre a descrição semântica do serviço *BoletimPorAerodromo.wsdl*, com as anotações semânticas de segurança inseridas.

Para a realização do experimento, foram utilizadas as seguintes ferramentas no ambiente de desenvolvimento:

- *Eclipse Modeling Tools*: Version: Luna (4.4.0) [30];
- *Plugin* WSDL2OWL: faz parte de uma API (Java) OWL-S *Composer* 3.0 [31]; e
- *Protégé* Versão 4.2.

A descrição semântica do *BoletimPorAerodromo.wsdl*, em OWL-S, foi realizada automaticamente utilizando o *plugin* WSDL2OWLS, integrado ao *Eclipse*, gerando a ontologia *BoletimPorAerodromo.owl.*

A Fig. 10 demonstra a descrição textual de *Service* do serviço *BoletimPorAerodromo.owl.*

```
<service:Service rdf:ID="BoletimPorAerodromoService">
  <service:presents>
    <profile:Profile rdf:ID="BoletimPorAerodromoProfile"/>
  </service:presents>
  <service:describedBy>
    <process:AtomicProcess rdf:ID="BoletimPorAerodromoProcess"/>
  </service:describedBy>
  <service:supports>
    <grounding:WsdlGrounding rdf:ID="BoletimPorAerodromoGrounding"/>
  </service:supports>
</service:Service>
```

Fig. 10. Descrição textual de *Service* de *BoletimPorAerodromo.owl*.

As anotações semânticas de segurança da ontologia *BoletimPorAerodromo.owl* foram inseridas, manualmente, utilizando o OWL-S Editor, integrado ao Eclipse. A Fig. 11 demonstra a descrição do *Profile* da ontologia gerada, sem as propriedades de segurança, enquanto a Fig. 12 apresenta a mesma com as propriedades já inseridas.

```
<!-- Profile description -->
<profile:Profile rdf:about="#BoletimPorAerodromoProfile">
  <service:presentedBy rdf:resource="BoletimPorAerodromoService"/>
  <profile:serviceName>BoletimPorAerodromo</profile:serviceName>
  <profile:textDescription>Serviço Web de Boletim Meteorológico de Aeródromo
   </profile:textDescription>
  <profile:hasInput rdf:resource="#AerodromoId"/>
  <profile:hasOutput rdf:resource="#boletim"/>
</profile:Profile>
```

Fig. 11. Profile de *BoletimPorAerodromo.owl* sem anotações.

```
<!-- Profile description -->
<profile:Profile rdf:about="#BoletimPorAerodromoProfile">
  <service:presentedBy rdf:resource="BoletimPorAerodromoService"/>
  <profile:serviceName>BoletimPorAerodromo</profile:serviceName>
  <profile:textDescription>Serviço Web de Boletim Meteorológico de Aeródromo
   </profile:textDescription>
  <profile:hasInput rdf:resource="#AerodromoId"/>
  <profile:hasOutput rdf:resource="#boletim"/>
  <serviceSecurity:securityRequirement rdf:resource="#Sec1"/>
  <serviceSecurity:securityRequirement rdf:resource="#Sec2"/>
  <serviceSecurity:securityCapability rdf:resource="#Sec3"/>
</profile:Profile>
```

Fig. 12. Profile de *BoletimPorAerodromo.owl* com anotações.

Após a descrição semântica, foi testada a ontologia através de consultas de *queries* SPARQL, utilizando-se a ferramenta *Protégé*, conforme demonstrado na Fig. 13.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX profile: <http://www.daml.org/services/owl-s/1.1/Profile.owl#>
PREFIX serviceSecurity: <http://www.daml.org/services/owl-s/security/serviceSecurity.owl#>
PREFIX security: <http://www.daml.org/services/owl-s/security/security.owl#>

SELECT *
        WHERE {  ?profile rdf:type profile:Profile .
                 ?profile profile:serviceName  ?Name_Servico .
                 ?profile profile:hasInput ?Entrada .
                 ?profile profile:hasOutput ?Saida  .
                 ?profile profile:textDescription ?Descricao  .
                 ?profile serviceSecurity:securityRequirement ?Requisito_Seguranca .
                 ?Requisito_Seguranca rdf:type security:KeyProtocol .
                 ?Requisito_Seguranca security:relSecNotation ?Seg_1 .
                 ?profile serviceSecurity:securityCapability ?Capacidade_Seguranca .
                 ?Capacidade_Seguranca rdf:type security:KeyProtocol .
                 ?Capacidade_Seguranca  security:relSecNotation ?Seg_2 . }
```

Fig. 13. Relação das *queries* SPARQL.

Conforme a Tabela I, o resultado apresentado no *Protégé* demonstrou o correto relacionamento entre os parâmetros e as propriedades definidas na ontologia do serviço *BoletimPorAerodromo.owl*.

TABELA I – Resultado das consultas SPARQL

Respostas às ***Queries SPARQL*** no ***Protégé***	
Profile	BoletimPorAerodromoProfile
Nome_Servico	BoletimPorAerodromo
Entrada	AerodromoId
Saida	boletim
Descricao	Serviço Web de Boletim Meteorológico de Aeródromo
Requisito_Seguranca	Sec1 Sec2
Capacidade_seguranca	Sec3
Sec1	Authentication
Sec2	Authorization
Sec3	Encryption

As características informadas na Tabela I poderiam ser utilizadas por ferramentas de *matching*, no processo de descoberta e seleção de serviços, possibilitando que os usuários verifiquem se, além das características funcionais desejadas, as condições de segurança atendem as suas necessidades.

VI. Considerações Finais

O conhecimento apresentado neste artigo poderá ser útil no processo de estudo e definição de conceitos e tecnologias para a implantação do SWIM, na modernização do Sistema de Controle do Espaço Aéreo Brasileiro, pois o uso da semântica e de ontologias na descrição de características funcionais e de segurança dos serviços, possibilitará maior eficiência no processo de descoberta e seleção dos mesmos, permitindo que usuários, em todos os níveis de decisão ou aplicação, acessem simultaneamente informações confiáveis de forma mais rápida, precisa e segura, o que poderá contribuir para o aumento da capacidade e segurança operacional do referido Sistema.

Como trabalhos futuros, seria importante abordar a definição e composição semântica de políticas de segurança complexas, que poderiam ser utilizadas como parâmetros de entrada no processo de descoberta e seleção de serviços Web.

REFERÊNCIAS

[1] OASIS. Advancing Open Standards for the Information Society - Reference Architecture Foundation for Service Oriented Architecture Version 1.0. Disponível: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.html. Acesso: 4 jun. 2015.

[2] Marzullo, F. P.. SOA na Prática: Inovando seu Negócio por Meio de Soluções Orientadas a Serviços. São Paulo: Novatec Ed., 2012. 2ed.

[3] Almeida, Johanlemberg Ferreira de. Um modelo de alinhamento de sistemas de comando e controle. 2009. 108f. Tese de mestrado em Informática – Instituto Tecnológico de Aeronáutica, São José dos Campos.

[4] Berners-Lee T, Hendler J, Lassila O. The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American. May 2001:34-43.

[5] Denker, G., Nguyen, S., & Ton, A. (2004). OWL-S Semantics of Security Web Services : a Case Study. Security, 240–253.

[6] Denker, G., Kagal, L., Finin, T., & Paolucci, M. (2003). Security for DAML Web Services : Annotation and Matchmaking. In The Semantic Web-ISWC 2003 (pp. 335–350). Springer.

[7] Xavier, Otávio C.. Serviços Web Semânticos Baseados em RESTful. Goiânia, 2011. 135p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

[8] International Civil Aviation Organization – ICAO (2012). 2013-2028 Global Air Navigation Capacity & Efficiency Plan (Doc 9750-AN/963 Fourth Edition – 2013). Available from Internet: http://www.icao.int/publications/Documents/9750_4ed_en.pdf.

[9] BRASIL. Comando da Aeronáutica. Departamento de Controle do Espaço Aéreo. SISTEMA DE CONTROLE DO ESPAÇO AÉREO BRASILEIRO. NSCA 351-1 [Rio de Janeiro], 2010.

[10] BRASIL. Departamento de Controle do Espaço Aéreo. Disponível: http://www.decea.gov.br/sirius/. Acessado: 21 jun. 2015.

[11] SWIM. http://www.sesarju.eu/sesar-solutions/. Acessado: 21 jun 15.

[12] NEXTGEN - SESAR Data Model Coordination Group (NSDMCG). ICAO Global Interoperability Framework (IGIF) considerations, (December). 2014. p. 1–14. Disponível: https://www.eurocontrol.int/sites/default/files/content/documents/information-management/NSDMCG_ICAO_Global_Interoperability_Framework_Considerations.pdf.

[13] Costa, I., Monteiro, G., Barreto, A. B., Oliveira, J.P. (2014). Uma Proposta de Arquitetura baseada no SWIM para a implementação do Airport-CDM, 119–124.

[14] W3C. Web Services Architecture - W3C Working Draft 8, August 2003. Disponível: http://www.w3.org/TR/ws-arch/. Acesso : 4 jun 15.

[15] Shadbolt, N., Hall, W., & Berners-Lee, T. (2006). The semantic web revisited. IEEE Intelligent Systems.

[16] Garcia, Diego Zuquim Guimarães,; Toledo, Maria Beatriz Felgar de. (2008). Web Service Management Using Semantic Web Techniques. Proceedings of the 2008 ACM Symposium on Applied Computing.

[17] Gruber, T. R. A Translation approach to portable ontology specifications. Knowledge Acquisition, v. 5, p.199-220, 1993. Disponível em: http://tomgruber.org/writing/ontolingua-kaj-1993.pdf Acesso: 5 jun. 2015.

[18] W3C. OWL: Web Ontology Language (OWL). Dez 2012. Disponível em: http://www.w3.org/2001/sw/wiki/OWL. Acesso: 4 jun. 2015.

[19] W3C. OWL-S: semantic markup for web services. Nov. 2004. Disponível: <http://www.w3.org/Submission/owl-s>. Acesso: 4 jun15.

[20] Kagal, L., Finin, T., Paolucci, M., Srinivasan, N., Sycara, K., & Denker, G. (2004). Authorization and privacy for semantic web service.

[21] WS-Security 1.1: http://www.oasis-open.org/specs/. Acesso: 4 jun 15.

[22] WS-SecurityPolicy: http://www.oasis-open.org/specs. Acesso:4jun 15.

[23] Brahim, M. Ben, Chaari, T., Jemaa, M. Ben, & Jmaiel, M. (2012). Semantic matching of Web services security policies.

[24] WS-Policy 1.5: http://www.w3.org/TR/ws-policy/. Acesso: 4 jun 15.

[25] Zheng-qiu, H., Li-fa, W., Zheng, H., & Hai-guang, L. (2009). Semantic security policy for web service. Proceedings - 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA 2009, 258–262. http://doi.org/10.1109/ISPA.2009.10.

[26] SESAR: http://www.sesarju.eu/swim/. Acesso: 17 Ago. 2015.

[27] Kagal, L. (2002). Rei:A Policy Language for the Me-Centric Project.

[28] SPARQL. http://www.w3.org/TR/rdf-sparql-query/. Acesso: 30 jun 2015.

[29] PROTÉGÉ: http://protege.stanford.edu/. Acesso: 30 jun. 2015.

[30] ECLIPSE: https://www.eclipse.org/. Acesso: 2 mar. 2015.

[31] https://github.com/FORMAS/OWL-S-Composer. Acesso: 2 mar. 15.