

Interoperabilidade para o Ministério da Defesa: Rumo ao Desenvolvimento de Uma Ontologia de Topo

Luis Antonio de Almeida Rodriguez¹, José Maria Parente de Oliveira¹

¹Instituto Tecnológico de Aeronáutica

Praça Marechal Eduardo Gomes, Vila das Acácias, CEP 12 228-900 – São José dos Campos - SP – Brazil

Resumo — A inexistência de uma ontologia de topo que descreva o domínio do Ministério da Defesa Brasileiro (MD) e que permita a troca de dados entre Sistemas de Informações é um problema. Essa lacuna de tecnologia existente cria uma barreira para operar os sistemas das Forças Armadas Brasileiras de forma integrada com aqueles empregados por coalizões militares, como a OTAN, que fazem uso da camada de ontologias em suas arquiteturas de interoperabilidade. Este artigo descreve o caminho que o MD deve tomar rumo ao desenvolvimento de uma Ontologia de Topo e, como prova de conceito, mostra um protótipo de Ontologia descrito em OWL que descreve uma porção pequena daquele Ministério, um artefato com poder computacional que pode ser efetivamente processado por Sistemas Especialistas, parsers da linguagem ou editores de ontologias. A representação obtida é sugerida como ponto de partida para o desenvolvimento colaborativo de um artefato que permita a representação do domínio com expressividade tal que reflita a realidade e que possa ser processável por computadores.

Palavras-Chave — Gestão de Conhecimento, Ontologias, Interoperabilidade.

I. INTRODUÇÃO

Seres humanos procuram modelos conceituais para representar e explicar a realidade que acontece à sua volta. Quando ocorre, de alguma forma, uma transmissão de conhecimento entre duas entidades, é necessário saber que tipo de mecanismo interpretador é usado por quem o recebe, para ter certeza de que a comunicação será compreendida. Para que sistemas computacionais auxiliem na transmissão desse conhecimento, faz-se necessário que os mesmos interpretem essas percepções, através das ontologias [10].

O termo ontologia deriva, historicamente, de dois termos gregos: - “*ontos*”, que significa ser e “*logos*”, que significa palavra [2]. Aristóteles em IV a.c. considerava o termo de forma a permitir que ele fosse usado para classificar alguma coisa, ou, qualquer coisa, e criou categorias que serviram de base para inserir entidades em taxonomias específicas, que são utilizadas pelas ciências biológicas até hoje.

A construção de ontologias vem despertando a atenção da comunidade científica no sentido de considerá-las, ao desenvolver sistemas de informações, [8] que sejam interoperáveis em seus domínios, permitindo a troca de informações entre sistemas afetos aos mesmos. Para isso, é necessário estabelecer a formalização e a conceitualização destes domínios, que podem ser alcançadas por meio de ontologias formais que os descrevam, algo que justifica a

importância significativa dada à pesquisa para descoberta de novas técnicas de estruturação e formalização das regras de negócio do domínio.

No sentido de padronizar a troca de informações, principalmente em operações que afetem a segurança nacional, o Departamento de Defesa Dos Estados Unidos criou o Global Information Grid (GIG), que consiste em uma infraestrutura segura e única, composta de várias redes de computadores e baseada na arquitetura SOA (Service Oriented Architecture) [22], cujos serviços são atualizados por Comunidades de Interesse que detêm o domínio de determinadas áreas de conhecimento.

A OTAN [21] seguindo os passos daquele Departamento de Defesa, criou seu Modelo de Interoperabilidade, que permite a troca de informações padronizada em Operações Conjuntas, entre as nações que fazem parte desta coalizão. A abordagem adotada é a utilização de ontologias na organização do conteúdo das fontes de dados, onde ela é criada por especialistas e define as regras que regulam a combinação entre termos e relações do domínio do conhecimento [7]. A OTAN define seu modelo de troca de dados através de um modelo de dados cujo nome é JC3IEDM [1].

Para concretizar esta ideia, a OTAN criou o NATO Network Enabled Capability (NNEC) [21] com o objetivo de fazer uso de todas as capacidades do país, detalhadas em todos os níveis, militar e civil, através de uma infraestrutura de informações padronizada de modo a formalizar a comunicação de dados entre aqueles níveis. O NNEC acredita que, caso haja uma mudança de cultura dos militares que permita maior interação entre eles e, conseqüentemente, maior nível de compartilhamento das informações em situações de emergências, catástrofes naturais, ou, até mesmo em guerras, é possível conseguir um melhor entendimento da consciência situacional em que se encontram, de forma a proporcionar uma tomada de decisão mais rápida [21].

Seguindo na mesma direção, o Ministério da Defesa Brasileiro (MD), diante da necessidade de integrar informações globais de forma automatizada e oriunda dos atuais Sistemas de Comando e Controle (C2) das Forças Armadas, criou o Grupo de Trabalho (GT) Interoperabilidade. O objetivo definido para este grupo é o de propor soluções para aumentar a interoperabilidade entre as FAs, por meio de sistemas de C2, nos níveis estratégico e operacional, tendo como premissas as diretrizes, o conteúdo doutrinário, os procedimentos e as metas estabelecidas nos documentos específicos do MD [1].

Fica clara, neste ponto do texto, a tendência mundial de padronizar a troca de informações entre sistemas militares a fim de atender às necessidades de C2 das coalizões. É possível perceber, então, que o problema de pesquisa deste trabalho é a inexistência de um Modelo de Representação Conceitual que descreva todo o Ministério da Defesa Brasileiro, que seja processado por computador e que permita troca de dados entre seus Sistemas de Informação.

Sendo assim, o objetivo dos autores é propor o caminho para o desenvolvimento de uma Ontologia de Topo que sirva de modelo de interoperabilidade para o Ministério da Defesa Brasileiro, fazendo uso de pessoas, uma metodologia de desenvolvimento e uma ferramenta de software que suporta o processo de codificação da ontologia. Para avaliar a consecução do objetivo, os autores consideraram a hipótese de gerar, ao final de todo o processo:

- Uma ontologia **“prova de conceito”** que descreva o domínio de um piloto militar voando, que pode ser de qualquer uma das três Forças Armadas, considerando somente o número necessário de entidades que ajudem a representar um piloto a bordo de uma aeronave em vôo.
- Essa ontologia deve estar persistida em disco, com a extensão “.owl”.
- Esse arquivo owl deve ser aberto por um editor de ontologias qualquer, sem perder as propriedades de quando foi construído.

Para tanto, este artigo está estruturado da seguinte forma: Na Seção 2 é feita uma revisão da literatura existente sobre o assunto. A Seção 3 apresenta, em detalhes, a proposta deste artigo, o desenvolvimento de uma ontologia, a seção 4 apresenta os resultados obtidos perante a validação do objetivo proposto e comparando com outro modelo, e a seção 5 mostra trabalhos futuros e as conclusões.

II. REVISÃO DA LITERATURA

Ontologias

Se for considerado o sentido filosófico do termo, ele é relativamente novo, [10] tendo sido introduzido pelas pesquisas da filosofia com o objetivo de estudar a distinção do ser como ele é. Desde o século XVII o termo é utilizado para denominar o ramo da metafísica que estuda tudo aquilo que existe [2].

Nicola Guarino [10] isola o significado de ontologia de acordo com diferentes interpretações:

- 1- Uma disciplina filosófica;
- 2- Um sistema conceitual informal;
- 3- A especificação de uma conceitualização;
- 4- Um vocabulário utilizado por uma teoria lógica.

Ontologias também são vistas como coleções de conhecimento estruturado [13]. Uma ontologia representa uma lógica que descreve um significado específico para todo um vocabulário formal, ou seja, ela representa um compromisso em conceitualizar um domínio específico do conhecimento humano. A importância de seu uso está em conseguir representar hierarquias de classes de objetos e seus

relacionamentos, apresentadas em estruturas que validam a comunicação e integração entre sistemas de software.

Desenvolvimento de Ontologias

O processo de construção de uma ontologia não é uma tarefa trivial, visto que, para a definição da mesma, é necessário possuir conhecimento específico do domínio, a fim de evitar ambiguidade [13] e permitir que a mesma seja validada para uso em qualquer fim dentro daquele mesmo domínio.

Apesar da pesquisa com desenvolvimento deste tipo de artefato vir sendo realizada há décadas, não há consenso da comunidade de pesquisadores sobre um método que se sobressaia perante os demais já publicados [3]. A falta de uma metodologia consagrada é um problema para os desenvolvedores, visto que a maioria dos projetos fracassam ou não têm aceitação do produto pelo cliente.

Em termos práticos, desenvolver uma ontologia consiste em : [14]

- Definir as Classes de objetos da ontologia
- Organizar essas classes, segundo uma hierarquia taxonômica (superclasse – subclasse)
- Definir intervalos e os valores permitidos para eles, e
- Preencher estes intervalos com instâncias dos objetos.

É possível criar o conhecimento base de um domínio definindo instâncias individuais das classes de objetos e adicionar, nos intervalos, restrições e informações adicionais.

Forças Armadas de vários países utilizam modelos de dados padronizados baseados em ontologias, para a realização de operações conjuntas (Países diferentes) e combinadas (Forças diferentes, como marinhas e forças aéreas).

O Método Ushold & King de Desenvolvimento de Ontologias

Segundo os autores desta metodologia [4], o processo de desenvolvimento de uma ontologia deve ser guiado por cenários motivadores. A Técnica dos Cenários foi primeiramente apresentada por Carrol [4] para ser utilizada em projetos de interfaces humano-computadores. Ela foi largamente utilizada em desenvolvimento de software como Casos de Uso, nos desenvolvimentos orientados a objetos, como User Stories, nas Práticas Ágeis de Desenvolvimento, ou, como motivo para haver diferentes ramificações, ou versões, de projeto [4].

Os Cenários são hoje largamente utilizado para comunicação, modelagem e validação da informação entre usuários comuns da mesma. A Tabela 1 mostra alguns benefícios do uso de Cenários em desenvolvimentos de produtos computacionais:

Tabela 1 – Benefícios dos Cenários

Facilidade de entendimento Escrita em linguagem natural Vocabulário do domínio do problema e não a linguagem usada por engenheiros	Compromisso entre Cliente e Desenvolvedor Organização da Informação Rastreabilidade de requisitos
------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------

Os cenários são usados normalmente para descrever funcionalidades dos Sistemas de Informações. No caso do desenvolvimento de ontologias, cenários motivacionais são utilizados para estabelecer questionamentos, ou tipos de questionamentos, que ajudem a determinar qual o propósito da ontologia. "Uma excelente maneira de obter uma imagem clara do escopo da ontologia é criar cenários detalhados que serão usados nas aplicações que vão fazer uso da camada de ontologias [4].

A Fig. 1 mostra o processo de construção da ontologia, que é composto de quatro estágios distintos, a saber: **1. Identificação**, **2. Construção**, **3. Avaliação** e **4. Documentação**, descritos da seguinte forma:

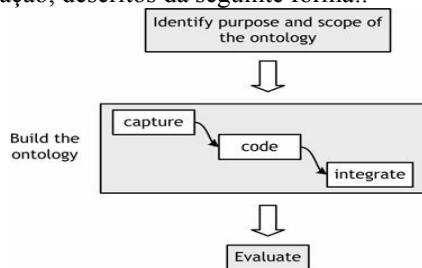


Fig. 1. – Método de Ushold e King [4]

O método foi bastante criticado por oferecer pouco suporte na identificação dos conceitos e relacionamentos, e, devido à elicitación de requisitos não ser formal, dificultando a percepção da rastreabilidade dos mesmos.

RapidOWL

Recentemente, o paradigma das Metodologias Ágeis vem ganhando muito sucesso sendo aplicado em desenvolvimento de Sistemas e de Softwares. Isso é devido ao fato de haver flexibilidade e capacidade de reações rápidas e eficientes face às mudanças constantes de requisitos, algo que é factual em engenharia de software e está incorporado às necessidades dos consumidores modernos de software.

Essa metodologia foi criada [16] para apoiar uma *Colaboração Semântica Adaptativa*, realizada entre especialistas de domínio e engenheiros de conhecimento. Ela é baseada na ideia do refinamento iterativo, marcação e estruturação de um conhecimento base. Ela foca na concentração sobre os menores blocos possíveis de informação que tenham um significado. O aspecto colaborativo pode ser entendido quando estes pedaços podem ser seletivamente unidos, removidos ou anotados com comentários.

A RapidOWL é inspirada na metodologia XP.K [16], como uma extensão da eXtreme Programming, e é destinada a ser uma proposta de metodologia ágil para desenvolvimento de sistemas baseados em conhecimento, conhecidos na literatura como *knowledge-based systems*. O método esgota a natureza genérica de um conhecimento base e focaliza no desenvolvimento de um conhecimento do qual não é possível conhecer o cenário em que será usado ou, o mesmo não pode ser definido.

Metodologias convencionais são atreladas a paradigmas como fases que são inseridas em um ciclo de vida

de desenvolvimento. As práticas ágeis acreditam que as mudanças devem ter um valor maior do que estar localizado em um ponto de um ciclo de vida. No lugar de propor um modelo de desenvolvimento, elas propõem valores cujos princípios são derivados dos processos gerais de engenharias diversas e, também, práticas necessárias para inserção da aplicabilidade desses princípios na vida diária, e esse é o caminho que torna possível descrever essa metodologia.

Os valores adotados por ela são semelhantes aos da XP (eXtreme Programming) [19]: Comunicação, Feedback, Simplicidade e coragem. A RapidOWL combina o primeiro e o segundo em um só chamado “Comunidade” e, além deste, inclui o valor da Transparência [16].

O processo de desenvolvimento é guiado por um meio-termo entre um conjunto de princípios, que inclui a premissa do escopo aberto, a promoção de mudanças incrementais, métodos de criação de uniformes, para tanto a modelagem, quanto para aquisição de instâncias, o desenvolvimento observável e feedback rápido. Este conjunto de princípios descreve a RapidOWL axiomáticamente, de um modo específico, que define as características desejadas para o processo. As práticas concretas são derivadas dos princípios que visem atingir as características desejadas na rotina diária, sem prescrição de um processo rígido.

As práticas da RapidOWL são inspiradas na eXtreme Programming e, devido às características específicas de engenharia do conhecimento, nem todas as práticas de XP tem uma equivalente em RapidOWL, o que também o é ,inversamente. Elas incluem, entre outras: Projeto Conjunto de Ontologia (para a colaboração entre os engenheiros do conhecimento, especialistas e usuários), Integração de Informações (para solidificar a elicitación do conhecimento sobre a informação existente), Geração de Visões (para fornecer visualizações de domínio específico para usuários humanos e sistemas de software) e a Evolução da Ontologia (permitindo a adoção harmoniosa das modelagens e correspondentes exemplos de migração de dados).

O autor a define como uma metodologia leve. Ela pode ser assim classificada porque a engenharia do conhecimento não é um negócio em si [16] (*negócio foi o termo usado aqui pela referência*) e, sendo assim, recursos significativos para a avaliação da metodologia para depois controlar o cumprimento dos processos não estão disponíveis. Ela focaliza na criação de ontologias independentes de domínio de aplicação, principalmente para serem utilizadas por tarefas de integração, e todas aquelas relacionadas ao desenvolvimento de conhecimento compartilhado, sistemas de classificação, vocabulários e conceitualizações.

III – PROPOSTA DE SOLUÇÃO

Modelos conceituais servem para descrever algo factível de existir no mundo real e refletem a percepção de quem os constrói. É necessário que as Instituições, Organizações e Empresas tenham suas Descrições Conceituais de Domínio a fim de que as mesmas conheçam suas essências e saibam como trocar dados entre seus parceiros e comunidades.

A ontologia desenvolvida neste trabalho tem como propósito representar o conhecimento de domínio do Ministério da Defesa Brasileiro, reunindo, em sua existência como prova de conceito, um pequeno conjunto de estereótipos existentes em um mundo real, que mostra a situação de uma Base Aérea usada, em situação de Guerra, para acomodar aeronaves e operações das três FFAA (Forças Armadas).

O processo de desenvolvimento da ontologia do MD

O desenvolvimento partiu da necessidade de se estabelecer um conjunto reduzido de termos, que coubesse dentro do escopo deste trabalho, uma prova de conceito. A necessidade inicial foi a de representar uma Base Aérea que contém aeronaves das três FFAA e todos os elementos envolvidos em uma situação de operação conjunta, onde aeronaves militares da Marinha, Exército e Força Aérea decolam e pousam a todo instante.

Os recursos humanos envolvidos na execução desse experimento foram duas pessoas, uma delas representando o Cliente, ou seja, quem vai fazer uso da ontologia e conhece o domínio e, a outra, o Desenvolvedor, que transformou as necessidades em requisitos a serem implementados na codificação da ontologia. Foi estabelecido um limite de tempo de três períodos de cinquenta minutos cada um, com intervalo de dez minutos entre eles, a fim de que fosse possível estabelecer um controle sobre o tempo do desenvolvimento.

A partir da necessidade de “*Representar uma Base Aérea*”, foi criado um conjunto de Questões de Relevância, que precisam ser respondidas para que os requisitos sejam validados, ou seja, as relações construídas e dispostas em triplas no artefato devem responder a estas perguntas para que a ontologia seja validada pelo cliente.

É possível observar que as questões existem para impor requisitos ao desenvolvimento: *A ontologia consegue representar uma aeronave de forma genérica? Ela consegue discriminar as aeronaves de cada força? Ela consegue representar uma relação entre piloto e aeronave? Ela consegue definir uma Identificação única para cada piloto e para cada aeronave? Ela descreve a estrutura hierárquica da Base Aérea, segundo o MD estabelece?*

Com um conjunto de Questões de Relevância estabelecido, o próximo passo realizado foi a redação de várias *User Stories* (US), artefatos oriundos de Práticas Ágeis de Desenvolvimento, como a SCRUM [18] e a XP [19]. Essas US servem para tornar explícitos os requisitos a serem atendidos no desenvolvimento da ontologia, em formato de histórias, resumidas a uma frase cada uma delas.

Essas histórias descrevem como o cliente e o desenvolvedor acham que o mundo real que está sendo representado funciona. Alguns exemplos dos requisitos especificados para esta ontologia: *Aeronave deve ser uma classe genérica e deve ter suas especializações para as três FFAA. À toda Aeronave em vôo deve estar associada uma instância da Classe Piloto. Um piloto só pode estar em um momento à bordo de uma aeronave. Aeronaves devem possuir um identificador único.*

Após estabelecido o conjunto de US que define o escopo do projeto, foi realizada uma retirada de todos os termos que continham significado para o domínio descrito. Palavras e expressões idiomáticas foram alocadas em uma Lista de Termos Relevantes (LTR), que reúne todas aquelas que têm um significado para a descrição do domínio desta Base Aérea, a qual pretende-se representar com a ontologia. A Tabela 2 mostra um pequeno conjunto dos termos gerados no processo e que fizeram parte da LTR que gerou a ontologia deste trabalho.

Tabela 2 – Lista de Termos Relevantes (LTR)

COMGAR	Aeronave	RealizaMissão	temFunção
GrandeComand	Tipo	DoTipo	temHorasVoadas
o	Missão	Piloto	HorasVoadas
FAE	Ataque	voa	Matrícula
FAEII	Escolta	Tripulante	LogBook
FAEIII	SAR	participaDeVo	estaVoandoEm
FFAA	Patrulha	o	temHorasTotais
FAB	TransporteTático	Trigrama	HorasTotais
EB	o	Patente	NomeBase
MARINHA	perenceA	FunçãoAbordo	

Esses termos foram transformados em classes de objetos existentes, nas propriedades e atributos dessas classes e nas propriedades de relacionamentos, que permitem que sejam construídas as triplas, que, juntas, formam o corpo da ontologia. Essa lista é armazenada em um arquivo comum de texto, que é absorvido pela ferramenta de software OntoWord (Seção III).

É importante salientar a necessidade de haver um especialista de domínio representando o cliente, tanto no processo de criação das User Stories, quanto da criação da LTR. A fidelidade com a realidade a ser representada depende do conhecimento de como as coisas existem em um cenário real, o uso do software proporciona automação na codificação, mas, a forma de representar o domínio depende do usuário.

A LTR é o objeto de entrada para o início da construção da ontologia utilizando o software de codificação OntoWord. O Desenvolvedor insere a LTR no ambiente gráfico do sistema e inicia a transformação dos termos elencados em construções nativas da ontologia. Primeiramente as Classes de objetos e Propriedades, logo após, a classificação vertical das mesmas, por heranças, e as formas com que as classes se atingem para relacionar-se. A Fig. 2 mostra a interface gráfica do sistema, após o usuário ter absorvido a LTR.

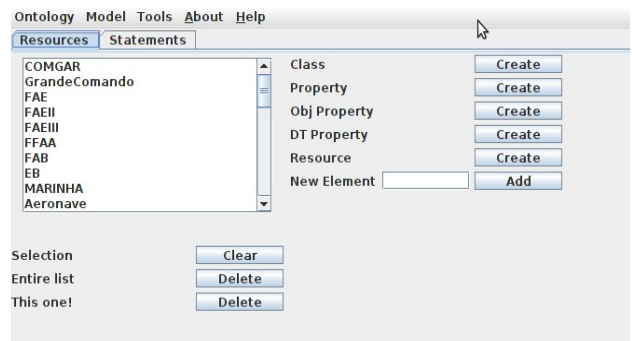


Fig. 2 – Interface gráfica do software OntoWord [17]

Após inserir os termos, o usuário pode iniciar a criação das construções nativas da ontologia através das funcionalidades localizadas na primeira aba da interface gráfica, como na Fig.2. A partir do momento em que estes botões são acionados, a aba de trás, de nome *Statements*, vai sendo preenchida automaticamente com o que foi criado pelo usuário na primeira aba, para que seja possível estabelecer os *statements*, que são formados pelas triplas – sujeito – predicado – objeto, que compõem a ontologia. A Fig.3 mostra um relacionamento, ou, *statement*, sendo criado.

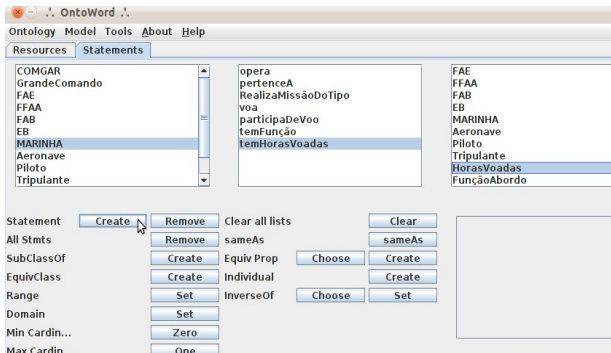


Fig. 3 – Criação de um *Statement* [17]

Após criados os *statements*, são inseridas, então, as instâncias das classes, ou seja, estas são populadas com os *Indivíduos* da ontologia. Nessa parte do processo, o *Cliente* e o *Desenvolvedor* podem inserir termos que não surgiram até aquele momento, mas, que podem complementar a descrição do conhecimento ali representado, dando mais fidelidade à realidade descrita.

Após realizadas as construções, a ontologia foi então persistida, a fim de que fosse verificado se as *Questões de Relevância* seriam respondidas. Caso não fossem, ocorreria o refatoramento, através do acréscimo de *Objetos* e relacionamentos que fizessem satisfeitas todas elas. A Fig. 4 mostra o arquivo *Teste1SIGE.owl* e sua codificação interna em linguagem owl [25].

Após a persistência, o mesmo arquivo foi aberto com o *Protegé 4.1*, onde foi possível perceber que nenhuma propriedade foi perdida e, ainda, a ontologia foi aberta com propriedades da *OWL Full*, um dialeto mais completo dessa linguagem. A Fig. 5 mostra uma visão pictorial, mais simpática à observação humana, da ontologia criada, utilizando o recurso da aba *OwlVIZ*.

On software de codificação de ontologias: OntoWord

É um aplicativo desenvolvido [11] na linguagem Java e destinado a semi-automatizar o processo de codificação de uma ontologia representada na linguagem Owl Lite [17]. Foi implementado usando o *Framework Jena* [5], que manipula os objetos de uma tripla RDF e possibilita criar conceitos e relações entre eles fazendo uso de componentes de interface gráfica para permitir que o usuário faça isso de forma visual.

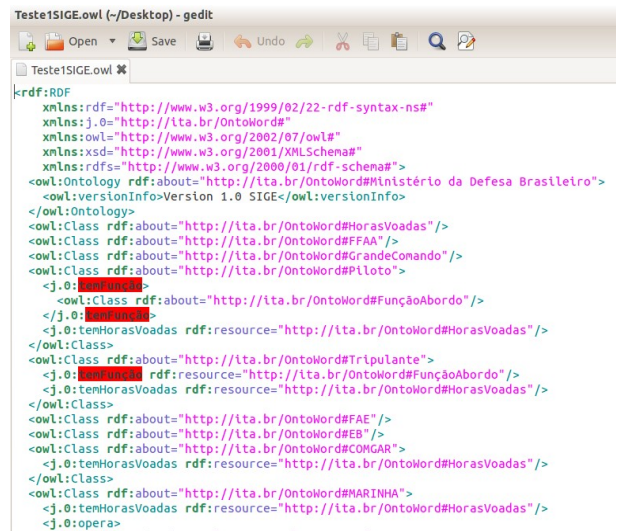


Fig. 4 – Ontologia do MD persistida em arquivo OWL

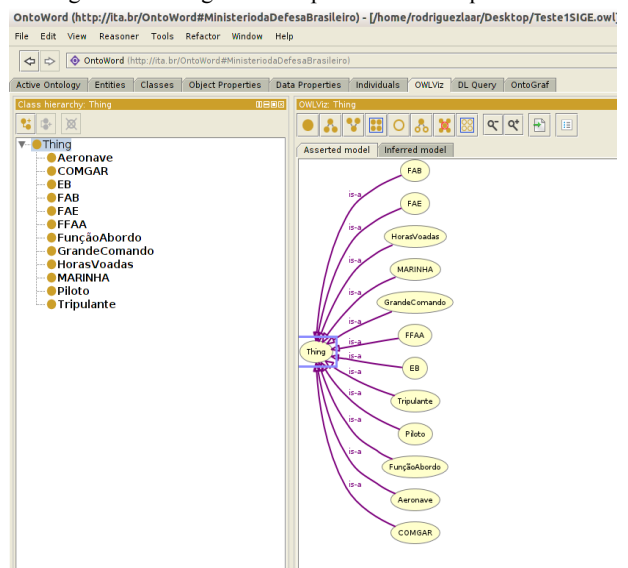


Fig. 5 – Ontologia do MD aberta pelo Protegé

A Fig.2 mostra a interface gráfica da ferramenta. Todas as funcionalidades do aplicativo são executadas dentro da janela principal, exceto o *help* e o *about*. As capacidades deste sistema são [17]:

- Importar uma lista de termos que fazem parte do domínio da ontologia e que serão usados para construir as triplas RDF
- Implementar as construções nativas da OWL Lite
- Persistir o modelo manipulado em disco em um arquivo owl que pode ser editado pelo Protegé 4.1 [15]

As capacidades desta ferramenta, se aliadas a um processo de engenharia de ontologias, podem torná-la um suporte valioso para automatizar a etapa de codificação.

Os termos localizados dentro dos componentes visuais da interface gráfica na Fig. 2 são o resultado da importação da lista de termos, que pode ser integrada com um

Super Sistema ou Standalone [17], através de um arquivo txt do HD. Esses termos são manipuláveis pelo usuário, a fim de que ele possa criar as construções nativas da owl e estabelecer relações entre os objetos da ontologia.

Na primeira aba interna do software é possível criar as OwlClasses, OwlProperties, OwlResources e acrescentar outros termos que não estavam presentes na LTR, que o cliente e o desenvolvedor julguem necessários. A partir do momento em que essas construções são criadas, elas ficam disponíveis na aba de trás do aplicativo, a fim de que possam ser selecionadas dentro do componente gráfico e associá-las, construindo os relacionamentos da ontologia.

A segunda aba da interface gráfica é destinada a relacionar as construções já criadas. É possível criar os *Statements* associando dois objetos das listas de componentes laterais, e uma propriedade, localizada na lista central da aba. A Fig. 2 mostra as construções sendo criadas na primeira aba e a Fig. 3 mostra os relacionamentos sendo criados, associando-se os elementos na segunda aba.

O ganho obtido com o uso dessa ferramenta está relacionado com a automação do processo de codificação, além do fato de que o uso da mesma é simples, se forem comparadas as quantidades de funcionalidades de cada um. O software, na verdade, pode ser colocado em uma categoria do Protegé, visto que absorve uma entrada para ser processada, algo que o famoso editor de *Stanford* não faz.

Aliada ao uso de boas práticas oriundas dos paradigmas de desenvolvimento ágil, torna-se possível desenvolver ontologias de forma colaborativa e com controle de escopo e tempo, conforme demonstrado na metodologia executada para o protótipo deste trabalho.

IV – RESULTADOS OBTIDOS

A ontologia desenvolvida neste trabalho descreve um pequeno espaço do domínio do Ministério da Defesa Brasileiro, contendo, para isso, um número reduzido de entidades que foram idealizadas com a finalidade de representar uma aeronave em voo pertencente a uma das três FFAA.

O artefato computacional gerado foi um arquivo de texto que contém uma ontologia owl persistida em disco. O arquivo pode ser aberto pelo editor de Ontologias Protegé 4.1 e manteve todas as propriedades que possuía quando foi criado pelo software OntoWord, sendo que, aberto por aquele editor, pode ser manipulado no formato OWL FULL [25], um dialeto mais completo da linguagem OWL.

O modelo produzido tem caráter conceitual e sugere-se que fosse desenvolvido seguindo este padoa, sendo que, o próximo passo para obter interoperabilidade com alguma coalizão militar, como a OTAN, é obter entendimento do modelo padronizado por essas organizações para que os sistemas de informação das FFAA, brasileiras que for operado de forma integrada com os de outras nações, ou, coalizões, aprenda a traduzir o tráfego da informação para enviar em seus formatos, receber e, traduzir novamente para a forma nativa.

A Fig. 6 mostra um pequeno trecho de um dos anexos da JC3IEDM [23] onde é descrita a entidade

AIRCRAFT-TYPE. É possível observar que trata-se da declaração de uma entidade com vários atributos, todos eles descritos de forma padronizada. Nota-se que o modelo da OTAN descreve, inclusive, a forma normal com deve ser implementado. Se esta entidade descrita for convertida em uma tabela de banco de dados, descrita em notação de Modelo Relacional, pode-se perceber as “colunas”, ou atributos, através da Fig 7.

A Fig. 8 mostra um trecho de um diagrama que define uma forma de representação que abrange todos os atributos definidos no modelo da OTAN, no formato de classes da UML [26], ela mostra as associações entre estes mesmos “objetos”, representados de um jeito pela OTAN e, de outro, mas tendo o mesmo conteúdo, pelas classes da ontologia desenvolvida neste trabalho.

Fig. 6 – Uma entidade descrita pela JC3IEDM [23]

AIRCRAFT-TYPE	An EQUIPMENT-TYPE that is designed to fly.	aircraft-type-id (PK) (FK) aircraft-type-category-code aircraft-type-airframe-design-code aircraft-type-model-code aircraft-type-manning-code aircraft-type-military-civilian-code aircraft-type-main-purpose-code aircraft-type-design-role-code aircraft-type-design-range-code aircraft-type-weather-qualifier-code aircraft-type-training-category-code aircraft-type-load-category-code aircraft-type-takeoff-and-landing-cc aircraft-type-wing-span-dimension
---------------	--------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

É possível observar na Fig. 8 que *AIRCRAFT-TYPE* é filha de outra classe. Há uma classe *AIRCRAFT*, que representa as aeronaves operadas por todas as FFAA, sendo que, a classe FFAA, tem três filhas, *EB*, *MAR* e *FAB*, que são as três Forças Armadas. Dessa forma, é possível especificar qual aeronave pertence a qual FFAA e obedecer ao paradigma da orientação a objetos, para relacionar as classes.

Fig. 7 – A mesma entidade representada em ModeloRelacional

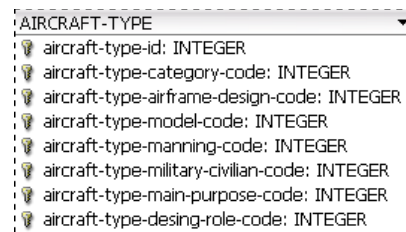
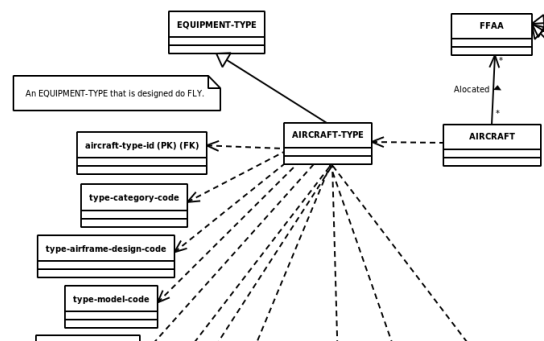


Fig. 8 – A entidade agora representada pelo diagrama de classes da UML 2



Para que haja interoperabilidade entre o modelo proposto por este trabalho e o modelo da OTAN, a Fig. 9 mostra um trecho do diagrama de classes da UML 2 do lado esquerdo e, do lado direito, a tabela que define a entidade do anexo [20]. O diagrama de classes mostra como a ontologia protótipo produzida considera a representação da mesma entidade e dos mesmos atributos do exemplo retirado da JC3IEDM.

Dessa forma, é possível ter o controle do uso de todos os tipos de entidades considerados pelo modelo MIP e seus atributos, sendo que o modelo produzido por este trabalho não considera adotar o mesmo tipo de estrutura determinado pelo modelo da OTAN, mas, sabe qual é esta estrutura e sabe trocar dados com ela, como é facilmente entendido na Fig. 9.

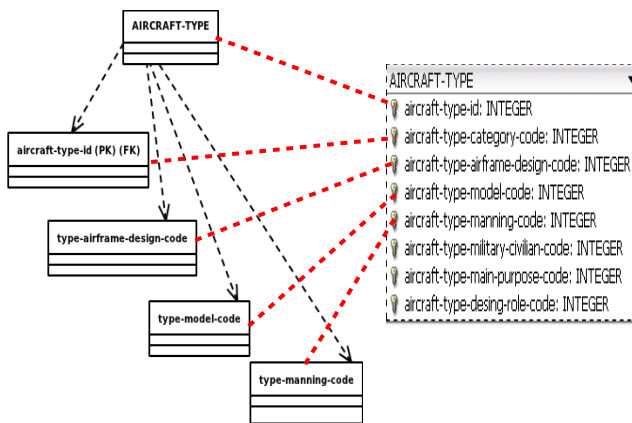


Fig. 9 – Correspondência entre as representações

O resultado obtido com este artefato produzido e a análise desta entidade do modelo da OTAN fortalece a crença da necessidade de criação de um modelo único de descrição de domínio para o MD. É possível perceber que se o modelo a se trocar dados for conhecido, e se o MD tiver um modelo próprio desenvolvido e utilizado pelos Sistemas de Informação Brasileiros, a tecnologia utilizada para a efetiva troca dos dados é conhecida e relativamente simples, já utilizada pela Força Aérea em desenvolvimentos internos.

O principal paradigma dos resultados foi o fato de ter-se conseguido representar o mesmo conteúdo do modelo da OTAN em Classes de uma ontologia, juntamente com seus relacionamentos.

V – TRABALHOS FUTUROS E CONCLUSÃO

Trabalhos Futuros

Conforme foi abordado na Seção III, o OntoWord deve ser considerado dentro de uma categoria diferente do Protegé, sendo assim, uma possibilidade de integração entre os dois sistemas seria interessante, visto que o segundo poderia absorver uma entrada e processá-la.

Isto fica sugerido visando não só a completude das ferramentas de software, mas, também, como sugestão de possível suporte a metodologias leves de desenvolvimento de ontologias.

Conclusão

É possível perceber, através da literatura aqui abordada, que as nações e suas FFAA estão em uma corrida pela padronização da troca de dados em Operações Militares. Adquirir a capacidade de interoperabilidade exige, nada mais, nada menos do que Sun Tzu já havia dito em seu livro “*A Arte da Guerra*”: Conhece a teu inimigo e a ti mesmo [24] ou seja, é preciso conhecer nossa organização para fazer com que ela possa se comunicar com outras, através de seus modelos, já existentes.

Para que o MD atinja os objetivos propostos ao GT de Interoperabilidade, é necessário que o domínio de funcionamento deste Ministério seja conhecido. Para tal, é necessário que ele seja representado computacionalmente, através de uma ontologia. Sem este tipo de artefato, não é possível entender o funcionamento da Organização e permitir que seus sistemas comuniquem-se, tanto interna, quanto externamente.

O desenvolvimento deste protótipo de ontologia foi realizado para servir como estímulo para que as FFAA e os órgãos que fazem parte do MD iniciem o desenvolvimento de suas ontologias de forma colaborativa, sugerindo uma metodologia leve e uma ferramenta de software eficaz na codificação do artefato computacional, permitindo um controle de escopo e tempo de desenvolvimento.

REFERÊNCIAS

- [1] ALMEIDA, J. F. **Um modelo de alinhamento de sistemas de comando e controle**. 2009. 108 f. Dissertação (Mestrado em Área de Informática) - Engenharia Eletrônica e Computação, Instituto Tecnológico de Aeronáutica, São José dos Campos.
- [2] ALMEIDA, M. B.; BAX, M. P. **Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção**. *Ci. Inf.* v. 32 n.3 Set./Dez. 2003. Access: <<http://www.scielo.br/pdf/ci/v32n3/19019.pdf>>.
- [3] BAUMEISTER, J. **An agile development of diagnostic knowledge systems**. 2004. 270 f. Dissertation (Erlangung des Naturwissenschaftlichen Doktorgrades für Mathematik und Informatik) - Bayerischen Julius-Maximilians-Universität Würzburg, Würzburg.
- [4] BREITMAN, K.K.; CASANOVA, M.A.; TRUSKOWSKI, W. **Semantic Web: Concepts, Technologies and Applications**. Ch. 8. 2007. NASA Monographs in Systems and Software Engineering. Free Books & Magazines. Access: <www.dbebooks.com>.
- [5] COSTA, A. D.; OLIVEIRA, J. M. P. **Transformando informação em conhecimento pela adição de semântica ao processo de classificação de documentos**. In: WORKSHOP ON ONTOLOGIES AND METAMODELING IN SOFTWARE AND DATA ENGINEERING, 3., 2008, Campinas. **Proceedings...**[S.l.: s.n], 2008.
- [6] DICKINSON, Ian. **JENA ontology API**. Tutorial para OWL API do Jena. Feb. 2009.. Access: <<http://Jena.sourceforge.net/ontology/>>.
- [7] GIL, Y.; RATNAKAR V. **A comparison of (semantic) markup languages**. Marina del Rey, CA: USC Information Sciences Institute and Computer Science, 2000. Access: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.5956&rep=rep1&type=pdf>>.
- [8] GÓMEZ-PÉREZ, A, FERNÁNDEZ-LÓPEZ, M.; VICENTE, A. **Towards a method to conceptualize domain ontologies**. In: WORKSHOP ON ONTOLOGICAL ENGINEERING. EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 12., Budapest, 1996. - **Proceedings...** [S.l.]: John Wiley & Sons, 1999. p. 41-52.

- [9] GRUBER, T. R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing In Guarino*, N. and Poli, R. (eds.), Formal Ontology in Conceptual Analysis and Knowledge Representation (to appear, 1995).
- [10] GUARINO, N. Formal ontology and information systems. In: FORMAL ONTOLOGY IN INFORMATION SYSTEMS, 1998, Trento. **Proceedings...** Amsterdam: IOS Press, 1998. p. 3- 15.
- [11] HEBELER, J. et al. **Semantic web programming**. Indianapolis: Wiley Publ., 2009. 651p. ISBN 978-0-470-41801-7.
- [12] LAZILHA, F.R.; BERNABÉ, F.S.; PASCUTTI, MCD. *Ontologias e bancos de dados. Iniciação Científica*, Maringá: CESUMAR, v. 1, n. 1, p. 43-47, 1999. Access: <<http://www.cesumar.br/pesquisa/periodicos/index.php/iccesumar/artic le/downlo ad/10/327>> .
- [13] LIBRELOTTO, G. R. **Representação de conhecimento na semantic web**. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMUNICAÇÃO, 15., 2005. **Anais...**[S.l.: s.n.], 2005.
- [14] NOY, N. F.; MCGUINNESS, D. L. **Ontology development 101: a guide to creating your first ontology**. Stanford, CA: Stanford University, 2001. Disponível em <http://protege.stanford.edu/publications/ontology_development/ontology101-noy-cguinness.html> .
- [15] PROTEGE. **Protégé Ontology Builder**. 2012. Access:<<http://www.stanford.edu/Protege>>.
- [16] RAPIDOWL. **RapidOWL - an Agile Knowledge Engineering Methodology** . Soren Auer , University of Leipzig , Institute for Computer Science , Augustusplatz 10-11, 04109 Leipzig, Germany . Access: <<http://http://www.informatik.uni-leipzig.de/~auer/publication/RapidOWL.pdf>>.
- [17] RODRIGUEZ, L.A.A. **OntoWord - Uma Ferramenta Para Construção de Ontologias a Partir de Histórias de Uso**. 2011. 120f. Trabalho de Conclusão de Curso. (Lato Senso) – Instituto Tecnológico de Aeronáutica, São José dos Campos.
- [18] SCRUM. Scrum.org: **The Home Of Scrum and your Source for Scrum training**. 2010. Access: <<http://www.scrum.org/>>.
- [19] XP, **eXtreme Programming. Extreme Programming: A gentle introduction**. Access: <<http://www.extremeprogramming.org>>.
- [20] MIP. The Multilateral Interoperability Programme. 2012. Access: <<http://mip-site.org/>>.
- [21] NATO. NATO Network Enabled Capability. 2012. Access: <http://www.nato.int/cps/en/natolive/topics_54644.htm>.
- [22] USA. UNITED STATES OF AMERICA. **Department of Defense**. Department of Defense Directive 8100.1. Global Information Grid Overarching Policy. 2002. Access: <<http://www.dtic.mil/whs/directives/corres/pdf/841002p.pdf>>.
- [23] JC3IEDM, Annex B – DMWG 20090514 Edition3.0.2 B-1. Access: <http://www.google.com/url?sa=t&rct=j&q=jc3iedm-annex%20b-entities-dmwg-edition_3.0.2_20090514&source=web&cd=2&ved=0C-CgQFjAB&url=https%3A%2F%2Fmipsite.lsec.dnd.ca%2Fpublic%2520Document%2520Library%2F99_Archives%2F03-Baseline_3.0%2FJC3IEDM-Joint_C3_Information_Exchange_Data_Model%2FJC3IEDM-Annex%2520B-Entities-DMWG-Edition_3.0.2_20090514.pdf&ei=3fo-SUKD0EL-g8QThwIDQCw&usq=AFQjCNELJCxfImKknB3wvSF-doiWeeqnWNw>.
- [24] TZU, S. Sun Tzu, The Art Of War., 2012. Access: <<http://www.artofwarsuntzu.com/Art%20of%20War%20PDF.pdf>>.
- [25] W3C. World Wide Web Consortium. 2012. Access: <<http://www.w3c.org>>.
- [26] UML. Unified Modelling Language. 2012. Access: <<http://www.uml.org/>>.