

OntoMap: De Mapas Conceituais a Ontologias OWL

Guilherme Nicchio Pinotte

Universidade Federal do
Espírito Santo

Av. Fernando Ferrari, 514
Goiabeiras, Vitória, ES, Brasil
+55 (27) 4009-2124

guilhermepinotte@gmail.com

Davidson Cury

Universidade Federal do
Espírito Santo

Av. Fernando Ferrari, 514
Goiabeiras, Vitória, ES, Brasil
+55 (27) 4009-2124

dedecury@gmail.com

Amal Zouaq

Royal Military College of Canada
CP 17000, Succursale Forces,
Kingston ON Canada K7K 7B4

+1-613-541-6000 # 6478
amal.zouaq@rmc.ca

ABSTRACT

Concept maps are graphical tools for organizing, representing and building knowledge. Recently, Knowledge Engineering has used Ontologies for knowledge management. Different researchers investigate the achievement of ontologies from the maps. This paper presents an architecture of a system designed to accomplish the transformation of a concept map in an OWL ontology. To this, it describes a proposal for mapping between Concept Maps and Ontologies. Visual aspects are also discussed and necessary adaptations to conventional concept maps editors in order to be able to support such mapping. In addition, a comparative analysis of existing web tools is provided, highlighting their strengths and weaknesses.

RESUMO

Mapas Conceituais são ferramentas gráficas para organização, representação e construção de conhecimento. Recentemente, a Engenharia do Conhecimento tem usado Ontologias para a gestão do conhecimento. Diferentes pesquisadores investigam a obtenção de ontologias a partir dos mapas. Este artigo apresenta uma arquitetura de um sistema que visa realizar a transformação de um mapa conceitual em uma ontologia OWL. Para tal, é descrita uma proposta de mapeamento entre os Mapas Conceituais e Ontologias. Além disso, são discutidos aspectos visuais e adaptações necessárias aos editores de Mapas Conceituais convencionais para que sejam capazes de suportar tal mapeamento. É também realizada uma análise comparativa de ferramentas já existentes na *web*, destacando suas vantagens e debilidades.

Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services – *Web-based services*.

K.3.0 [Computers and Education]: General.

I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods – *Representation languages*.

D.2.11 [Software Engineering]: Software Architecture – *Patterns*.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference'10, Month 1–2, 2010, City, State, Country.
Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

General Terms

Algorithms.

Keywords

Concept maps, web service platform, ontologies, knowledge representation.

1. INTRODUÇÃO

O tema Engenharia do Conhecimento está em evidência nos últimos anos. Nesse contexto, existem diferentes formas de representação de conhecimento, como os Mapas Conceituais e as Ontologias, especialmente no contexto da Web Semântica, onde idealizam-se ontologias processáveis por máquinas. Por um lado, os mapas conceituais são informais, simples de construir e de fácil entendimento. Por outro, as ontologias são formais e difíceis de criar, exigindo, portanto, a presença de um especialista em Engenharia do Conhecimento para criá-las.

Mapas Conceituais são ferramentas gráficas para organização, representação e construção de conhecimento. Eles são formados por conceitos, geralmente representados por círculos ou caixas de algum tipo, uma seta para ligar dois ou mais conceitos e um rótulo para definir a natureza da relação [19]. Já uma Ontologia é uma especificação explícita e formal de uma conceitualização compartilhada [25]. Em Guarino [12], são encontradas maiores informações sobre essa definição.

Existem várias linguagens de modelagem capazes de representar um conhecimento sobre um determinado domínio, onde cada uma delas possui seus próprios objetivos e níveis de complexidade na criação de seus modelos. Algumas dessas linguagens podem ser mais expressivas, e.g. UML [26], OntoUML [13] e OWL [21]. Já outras são menos expressivas, e.g. ER [5] e os Mapas Conceituais [20]. Importante ressaltar que, apesar de linguagens mais expressivas poderem representar conhecimento de forma mais clara e com boa fidelidade ao domínio, nem sempre seu uso é mais adequado quando comparado a outra menos expressiva. Junto à expressividade vem a complexidade, o que significa dizer que quanto mais expressiva, mais complexo é o desenvolvimento de modelos, além de aumentar a complexidade computacional [3]. Tal fato dificulta a construção de modelos por especialistas de domínio, que possivelmente não possuem conhecimento adequado ao uso de linguagens de modelagem mais complexas, como por exemplo, a OntoUML ou a OWL.

As Figuras 1, 2 e 3 representam um mesmo domínio descrito em diferentes linguagens de modelagem, evidenciando seus diferentes propósitos e níveis de complexidade na construção de modelos. Ao analisar em específico as Figuras 1 e 2, que mostram

linguagens tipicamente ontológicas, fica evidente a dificuldade que um especialista de um domínio qualquer teria na representação de conhecimento. Em contrapartida, a Figura 3, representada por um mapa conceitual, deixa claras a simplicidade e informalidade da linguagem.

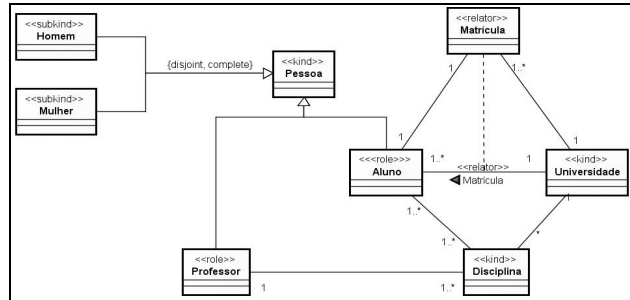


Figura 1. Exemplo de um domínio representado em OntoUML.

```

12 <Ontology xmlns="http://www.w3.org/2002/07/owl#"
13 <Declaration>
14 <Class IRI="#Professor"/>
15 </Declaration>
16 <Declaration>
17 <Class IRI="#Pessoa"/>
18 </Declaration>
19 <Declaration>
20 <Class IRI="#Disciplina"/>
21 </Declaration>
22 <SubClassOf>
23 <Class IRI="#Professor"/>
24 <Class IRI="#Pessoa"/>
25 </SubClassOf>
26 <ObjectPropertyDomain>
27 <ObjectProperty IRI="#lecionaEm"/>
28 <Class IRI="#Professor"/>
29 </ObjectPropertyDomain>
30 <ObjectPropertyRange>
31 <ObjectProperty IRI="#lecionaEm"/>
32 <Class IRI="#Disciplina"/>
33 </ObjectPropertyRange>
34 </Ontology>
    
```

Figura 2. Trecho de código de um domínio representado em OWL.

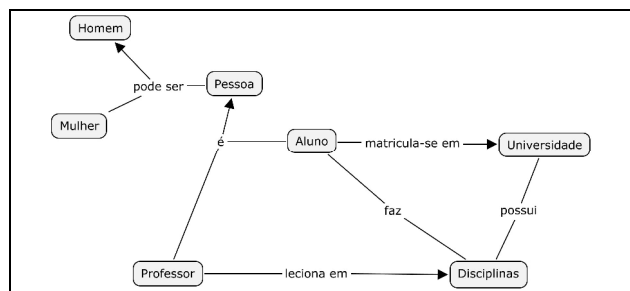


Figura 3. Exemplo de um domínio representado em mapa conceitual.

Dado o exposto, este trabalho propõe, por meio dos mapas conceituais, fornecer aos não especialistas em Engenharia do Conhecimento a possibilidade de representar conhecimento de um domínio qualquer, de maneira simples e sem a necessidade de recorrer a um especialista em computação. Em outras palavras, este trabalho tem o objetivo de transformar o conhecimento informal descrito por mapas conceituais em conhecimento formal

das ontologias rasas descritas na linguagem OWL, que é o padrão adotado pela Web Semântica.

Para tal, apresentamos uma proposta da arquitetura de um sistema que estamos chamando de OntoMap. O OntoMap é uma ferramenta idealizada com o propósito de transformar mapas conceituais em ontologias OWL. A ferramenta é o resultado da interação de duas partes básicas: um editor de mapas conceituais adaptado que poderá ser acessado por meio de uma interface de uso, que estamos chamando de nosso Portal [6], e um serviço que contém a lógica de mapeamento dos mapas conceituais para as ontologias OWL, que será hospedado em uma plataforma web orientada a serviços, chamada CMPaaS [7]. Esta plataforma é um projeto maior que abriga vários serviços sobre mapas conceituais da qual iremos falar sumariamente na Seção 2 deste trabalho.

Este artigo está estruturado da seguinte forma: A Seção 2 apresenta o contexto do nosso editor de mapas e ontologias. A Seção 3 apresenta uma revisão literária do estado da arte e da tecnologia. A Seção 4 descreve a arquitetura proposta explicando cada componente da mesma. A Seção 5 exhibe um primeiro protótipo como prova da validade conceitual do projeto. Por fim, a Seção 6 apresenta algumas conclusões temporárias bem como aponta para alguns possíveis trabalhos futuros.

2. SOBRE O CONTEXTO

Exploramos há cerca de 15 anos o domínio do conhecimento mapas conceituais aplicado aos processos da aprendizagem. Durante este tempo orientamos inúmeros projetos de graduação e mestrado e vários de doutorado sobre o tema. Cada um desses trabalhos gerou ferramentas prototípicas que evoluíram ao longo do tempo, algumas delas já com o status de ferramenta em produção. No entanto, enquanto ferramentas isoladas, são pouco eficientes. Logo, decidimos por integrá-las. Disto, falaremos um pouco a seguir.

2.1 Dos mapas na aprendizagem

Os mapas conceituais foram pensados e desenvolvidos por Novak [20] que os definiu como uma ferramenta para representar e organizar o conhecimento. Um mapa é uma unidade gráfica a duas dimensões de conceitos construídos de tal forma que as relações estabelecidas entre eles sejam evidentes. Consideramos que em um mapa, os conceitos são substantivos, representados por retângulos, e os rótulos das ligações são estruturas verbais. A tripla (conceito, ligação, conceito) forma uma unidade semântica chamada proposição, onde pode-se considerar que uma proposição é a menor unidade de conhecimento dentro de um mapa. As proposições constituem a característica básica dos mapas conceituais, o que os distingue de outras representações semelhantes [18]. Novak apoia suas definições na Teoria da Aprendizagem Significativa de Ausubel [2]. De acordo com essa teoria, de natureza cognitivista, uma estrutura mental do conhecimento se organiza de maneira hierárquica, de forma arborescente, onde conceitos mais genéricos estão nos seus níveis mais elevados, próximos à raiz, enquanto conceitos mais específicos figuram em níveis mais baixos se alongando até suas folhas. Em Ausubel [1], conceitos e proposições são os blocos de construção do conhecimento. Ele define subsunção como sendo uma operação que se caracteriza por: classificar, incorporar ou incluir algo em uma categoria ou em um princípio mais geral. Nessa teoria os conceitos subsunçores são conceitos mais gerais, e já estáveis, que figuram na estrutura cognitiva de um indivíduo e que se prestam à ancoragem de novos conceitos mais específicos.

em grupos. Quando aplicados na aprendizagem colaborativa, eles permitem:

- estudantes, ambos, individualmente e como um grupo, observar e analisar a articulação, organização e avaliação crítica de todo o processo de construção do conhecimento;
- professores monitorar e avaliar o progresso do conhecimento de cada estudante e de um grupo de trabalho, observando os seus mapas.

Comunidade Global

- Os mapas permitem a gravação de reuniões em atas.
- Os mapas permitem a indexação de conceitos, dentre outras tarefas.
- Os mapas podem substituir organogramas razoavelmente bem, permitindo que ambos, para a estruturação de uma equipe, bem como o estabelecimento de subordinação e de responsabilidade para cada um dos membros.

Engenharia do Conhecimento

- Os mapas permitem a representação e organização do conhecimento de uma companhia ou qualquer outra instituição.
- Os mapas permitem a modelagem dos processos de uma empresa ou organização;
- Os mapas viabilizam as memórias organizacionais;

Destacamos ainda que, aplicações de mapas conceituais na educação infantil são adequadas para crianças com dificuldades na construção de frases. Além disso, tendo em vista que os mapas têm potencial lúdico, eles incentivam e facilitam que as crianças expressem suas ideias.

2.2 Ontologias a partir de Mapas: Por quê?

Recentemente a Engenharia do Conhecimento tem usado ontologias para a gestão do conhecimento. A gestão do conhecimento está, funcionalmente, envolvida na geração, armazenamento e disseminação do conhecimento dentro de qualquer tipo de corporação. Ela engloba diversas atividades, que vão desde técnicas para recursos humanos até técnicas relacionadas com sistemas de informação [15].

As ontologias, por outro lado, para serem manipuladas por computadores, têm sido representadas, principalmente, em lógica descritiva. Elas têm jogado um importante papel em incontáveis atividades, principalmente na gestão do conhecimento com respeito à construção de interfaces humano-máquina intuitivas, recuperação inteligente de informação e, na Web Semântica, entre outras atividades. Ontologias têm sido também usadas na captura de conhecimento sobre algum domínio de interesse.

Por que usar mapas conceituais no contexto da gestão do conhecimento? Porque os mapas conceituais são considerados uma ferramenta bem sucedida para compartilhar o conhecimento de um determinado domínio. Além disso, eles são bons para comunicar consenso e para ajudar a comunicação entre especialistas de domínio e engenheiros de conhecimento.

Mapas conceituais e ontologias são ferramentas muito similares, especialmente ao se analisar sua estrutura. Portanto, nós

acreditamos que é possível a criação de ontologias rasas a partir dos mapas. No entanto, os mapas conceituais não sem exigem o rígido formalismo das ontologias. Devido suas topologias gráficas, ambos podem ser facilmente processados por computadores. Além disso, suas estruturas proposicionais são muito semelhantes à estrutura utilizada para representar as propriedades em lógica descritiva. Os autores em [24], [29], [11], entre outros, sugeriram a criação de um procedimento para apoiar a transformação de mapas conceituais em uma base de conhecimento representado em lógica descritiva.

Para este trabalho estamos interessados em ontologias rasas como representações de conhecimento especialmente em apoio à aprendizagem e na gestão do conhecimento. Usando a arquitetura baseada em agentes, ontologias também podem orientar a construção de ambientes virtuais de apoio à aprendizagem e modelagem cognitiva de estudantes, considerando suas produções individuais e aquelas resultantes da cooperação e colaboração. Ontologias também podem ser úteis no apoio à construção de textos gramaticalmente corretos.

2.3 Sobre nossa plataforma

Ao lidar com o desenvolvimento de soluções computacionais, o assunto em voga nos últimos anos é, sem dúvida, computação em nuvem. Este é um modelo de computação no qual processamento, armazenamento e soluções computacionais (*software*) são oferecidos por um provedor de serviços acessados remotamente via internet. Esta tecnologia permite que aplicações possam realizar a recuperação de informações de qualquer lugar, a partir de qualquer plataforma, em qualquer momento, utilizando apenas web em vez de aplicativos instalados localmente.

A principal vantagem oferecida pela computação em nuvem é a sua capacidade de seus serviços serem facilmente estendidos e incorporados em diversos aplicativos, aumentando assim a produtividade na criação de novas aplicações. Devido a esse fato, grandes companhias do ramo de tecnologia da informação (por exemplo, Facebook, Apple, Google, Twitter etc.) disponibilizam suas APIs (*Application Programming Interfaces*) de acesso aos seus serviços utilizando esse modelo de computação. Só para citar um exemplo, um aplicativo em nuvem amplamente utilizado é o Google Maps.

Hoje em dia, existem inúmeras aplicações que estendem suas funcionalidades oferecendo serviços complementares, tais como aplicações de geolocalização que controlam o percurso, ritmo e calorias consumidas por um atleta em uma atividade física. Para esse projeto, buscamos explorar essa capacidade de expansão e produtividade. Estamos criando a serviços básicos de edição, gerenciamento e manipulação de mapas conceituais que estarão disponíveis para qualquer pessoa no mundo por meio da nossa plataforma de serviços.

Uma das características fundamentais da arquitetura de *software* a ser utilizada neste projeto, conhecida como SOA (*Service Oriented Architecture*), é sua capacidade para promover a integração. Isto significa que novos serviços que estendem a funcionalidade dos serviços oferecidos por nossa plataforma podem ser desenvolvidos e disponibilizados por qualquer pessoa e em qualquer lugar do mundo.

Até o presente momento, essa plataforma oferece os seguintes serviços:

- geração automática de mapas a partir de textos não estruturados,
- mesclagem de mapas [27],
- comparação de mapas,
- editor de mapas,
- correção de mapas,
- recuperação de informação a partir de perguntas em linguagem natural [22].

A Seção 4 apresenta maiores detalhes sobre a arquitetura geral de nosso projeto como um todo, destacando suas duas principais camadas: a de Visão e a de Serviços.

3. REVISÃO DA LITERATURA DAS FERRAMENTAS DE MAPEAMENTO

O primeiro passo para construção deste trabalho foi a leitura de artigos e a realização de testes nas ferramentas disponíveis para *download* na *web*, a fim de descobrir suas características bem como suas vantagens e debilidades. Apresentamos a seguir, uma breve explanação dessas ferramentas.

O COE [14], acrônimo para *Concept-Map Ontology Environment*, é uma ferramenta criada pelo IHMC (Florida Institute for Human & Machine Cognition) para construir, compartilhar e visualizar ontologias OWL. O COE foi construído baseado no CMapTools [4], que é um editor de mapas conceituais usado em ambientes educacionais e empresariais.

A ferramenta funciona da seguinte forma: ao criar uma nova relação entre dois conceitos, uma lista de estereótipos pré-definidos é exibida para o usuário. Caso deseje utilizar alguns dos estereótipos, o usuário apenas seleciona o mesmo dando dois “cliques” e a nova proposição é criada. Caso não utilize, o usuário constrói seu mapa normalmente, apenas escrevendo o rótulo da nova relação criada. Importante destacar dois pontos: 1) caso o usuário utilize um estereótipo pré-definido, tal como “*is a*”, esse mesmo corresponde a um *constructor* da linguagem OWL, o que permite que o algoritmo de mapeamento possa criar uma ontologia rasa mais axiomatizada e com maior fidelidade ao domínio em questão; 2) caso não utilize um estereótipo, esse novo rótulo criado pelo usuário é acrescentado à lista de estereótipos pré-existentes. Ou seja, ao criar outra relação o usuário pode utilizar o mesmo rótulo criado anteriormente apenas clicando no mesmo. Ao término da criação de um mapa, o usuário pode exportar o mesmo em formatos de linguagens ontológicas, como RDF e OWL. A Figura 5 mostra uma imagem do editor do COE, evidenciando a lista de estereótipos que o usuário pode selecionar para axiomatizar uma relação.

A ferramenta chamada MAP2OWL [10] é um *plugin* para o Protégé. O Protégé é um editor *open-source* de ontologia e um *framework* para construção de sistemas inteligentes. Dessa forma, a ferramenta MAP2OWL pode ser acessada pelo uso do Protégé. A dinâmica de funcionamento da ferramenta é a seguinte: o usuário cria um mapa conceitual da forma convencional e ao selecionar uma relação ou conceito, o usuário pode realizar a axiomatização de seu mapa por meio de um menu à esquerda do editor. Com uma relação selecionada, por exemplo, o usuário clica nas opções que o menu oferece para axiomatizar a mesma. Destacamos que para uma boa representação de conhecimento pelo uso dessa ferramenta, o usuário precisa conhecer sobre

ontologias OWL, pois é necessário saber o significado de palavras como “*functional*”, “*symmetric*”, dentre outras.

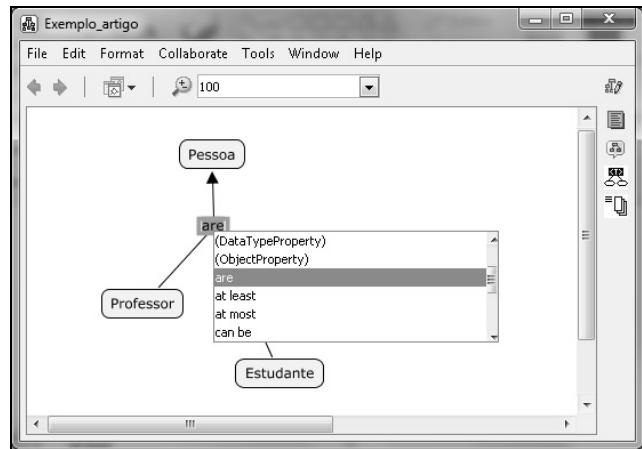


Figura 5. Editor do COE.

Como o MAP2OWL trabalha nativamente com OWL, a qualquer momento o usuário pode exportar o mapa conceitual em código OWL. Importante destacar as abas na parte superior do editor, com as quais o usuário tem o poder de decidir se deseja exibir elementos como classes e indivíduos no mapa.

A Figura 6 ilustra o editor do MAP2OWL, com seu menu de opções onde o usuário pode escolher ao clicar em uma relação.

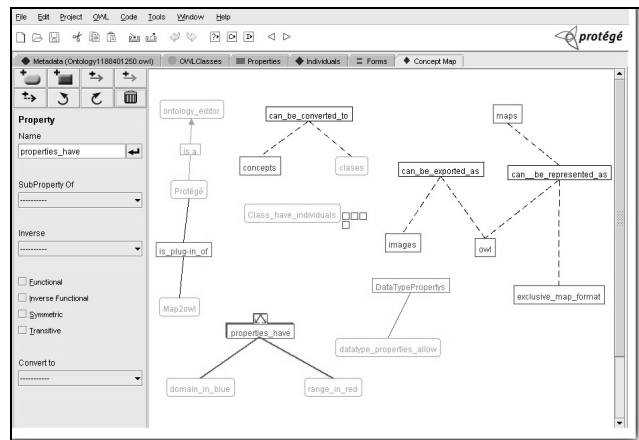


Figura 6. Editor do MAP2OWL. (Fonte: [10])

Em [23] é apresentada uma abordagem diferente das apresentadas anteriormente, na qual utiliza-se o processamento de linguagem natural para realizar o mapeamento entre os mapas conceituais e as ontologias. Os mapas conceituais são salvos em arquivos textos e o processamento de linguagem natural utiliza o WordNet. O processo de inferência é realizado após o processamento de linguagem natural, e assim, consegue-se detectar, classes, indivíduos, atributos e relações. O algoritmo é realizado apenas em Espanhol e não tivemos acesso ao editor para a realização de testes.

Apresentamos, a seguir, a Tabela 1 que contém uma síntese do que consideramos as vantagens e debilidades dessas ferramentas.

Tabela 1. Comparativo entre as ferramentas apresentadas.

Ferramenta	Vantagens	Debilidades
COE [14]	<ul style="list-style-type: none"> - mapeamento realizado através de estereótipos pré-definidos; - ambiente estável e disponível para <i>download</i>; - editor evidencia a diferença entre classes, indivíduos e atributos; - construção do mapa de forma simples, onde o usuário não é obrigado a escolher os estereótipos das relações. 	<ul style="list-style-type: none"> - o editor não muda automaticamente o layout de elementos distintos, como classes e indivíduos; - o usuário não dispõe de um menu no qual possa selecionar se deseja mostrar os elementos distintos da OWL, tais como classes, indivíduos e atributos, que torna a visão do editor poluída (com muitas informações); - o usuário não sabe o que cada estereótipo representa na linguagem OWL.
MAP2OWL [10]	<ul style="list-style-type: none"> - trabalha com OWL nativamente, ou seja, não realiza algoritmo de transformação entre mapas conceituais e OWL; - o editor evidencia a diferença entre classes, indivíduos e atributos; - o editor permite que o usuário selecione se deseja exibir diferentes elementos da OWL, como classes, indivíduos e atributos; 	<ul style="list-style-type: none"> - Por ser <i>plugin</i> do Protégé, a ferramenta fica limitada ao uso do Protégé; - Só funciona na versão 3.3 do Protégé; - Para uma boa representação de conhecimento, o usuário precisa saber o significado de palavras como <i>functional</i>, <i>symmetric</i>, <i>transitive</i>, dentre outras;
(sem nome) [23]	<ul style="list-style-type: none"> - mapeamento é feito através de processamento de linguagem natural; - o usuário cria o mapa conceitual sem ter conhecimento algum em OWL; - o editor de mapas conceituais não precisa de nenhum tipo de adaptação para realizar o mapeamento. 	<ul style="list-style-type: none"> - ferramenta só funciona na língua espanhola; - algoritmo depende do WordNet em Espanhol; - pelo mapeamento ocorrer através de processamento de linguagem natural, as regras de inferência são limitadas.

3.1 Conclusões da Seção

Sinalizamos que as ferramentas apresentadas nessa seção serviram de base para a construção de uma arquitetura conceitual do sistema OntoMap (Seção 4), além de um protótipo (Seção 5) que serão explicados em maiores detalhes nas seções seguintes.

Salientamos que tivemos acesso apenas aos editores do COE e MAP2OWL, onde foi possível a realização de testes nos mesmos. Para nosso protótipo, buscamos manter as principais vantagens que encontramos dessas ferramentas analisadas e eliminamos o que consideramos como fraquezas ou debilidades. Para nosso editor de mapas, unimos características dessas duas ferramentas. A seguir listamos as características que decidimos por manter em nosso editor:

- Lista de estereótipos pré-definidos. Esses estereótipos, tais como “*is a*”, “*same as*”, dentre outros, ficam disponíveis para o usuário ao criar uma nova relação. Característica proveniente do COE;
- Abas de seleção para exibição dos elementos distintos do OWL tais como classes, indivíduos e atributos. Dessa forma, fica a cargo do usuário decidir quais elementos ele deseja exibir na tela. Característica proveniente do MAP2OWL;
- Diferenciar visualmente os diferentes elementos da OWL. Esses elementos são: classes, indivíduos e atributos. Para isso, tomamos como base o projeto VOWL [28]. Essa característica foi encontrada tanto no COE quanto no MAP2OWL;

Com essas características, acreditamos que fomos capazes de manter a simplicidade do editor de mapas conceituais, sem poluir visualmente a visão do usuário.

4. DA ARQUITETURA E TECNOLOGIAS

A arquitetura proposta é o resultado da análise crítica que fizemos de algumas ferramentas, mostrada na seção anterior. Ela, no entanto, difere-se das analisadas por fazer parte de uma ampla plataforma de serviços baseada na arquitetura SOA (*Service Oriented Architecture*), na qual o OntoMap, terá seu serviço hospedado. Por ser uma plataforma *open-source*, além do serviço do OntoMap poder ser utilizado pela comunidade, ele poderá ser estendido e melhorado por aplicações clientes. Além disso, o editor da ferramenta OntoMap poderá ser acessado via nosso Portal, que é um sistema *web*, necessitando apenas de uma conexão com a internet.

Ainda se tratando de inovações que o OntoMap carrega, pode-se dizer que o objetivo dessa ferramenta é a construção de ontologias rasas que possam ser compartilhadas na *web*. Dessa forma, idealizamos que nossa ferramenta seja um provedor de informações, no caso, um provedor de ontologias. Um banco de ontologias será disponibilizado por meio da ligação com o DBpedia, fazendo parte do Linking Open Data [17]. O DBpedia é um esforço da comunidade *open-source* para extrair informações estruturadas do Wikipedia e disponibilizá-las na *web*. Além disso, é possível fazer consultas sofisticadas no Wikipedia e também ligar diferentes conjuntos de dados na *web* com os já presentes no site [8].

Dessa forma, idealizamos que uma ontologia rasa criada em nossa plataforma faça parte de algo maior. Tendo em vista a Figura 7, pode-se perceber que o DBpedia, localizado no centro da figura, é apenas um *dataset* dentre vários outros que estão ligados a ele. Ou seja, ligar nosso banco de ontologias com o DBpedia significa

ligar nosso *dataset* com outros *datasets* espalhados pelo mundo, fazendo parte do Linking Open Data.

Como dissemos, o OntoMap é a união de um serviço *web*, hospedado em nossa plataforma, com um editor adaptado de mapas conceituais, acessado pelo uso do nosso Portal. Portanto, podemos dizer que nosso Portal é a camada de visão, e a plataforma propriamente é a camada de serviços. A Figura 8 mostra a arquitetura sob uma perspectiva macro das duas camadas, onde as setas indicam as aplicações do Portal acessando serviços oferecidos na camada de serviços. Percebe-se que uma aplicação na camada de visão pode acessar mais de um serviço na plataforma, e um serviço na plataforma pode servir para mais de uma aplicação no Portal. Como já destacado anteriormente, os serviços hospedados em nossa plataforma serão disponibilizados para o uso de toda a comunidade. Sendo assim, é possível que outra camada de visão possa consumi-los e usá-los como bem entender.

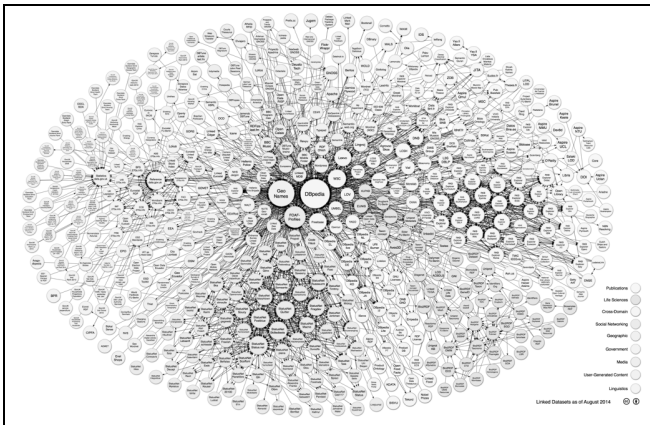


Figura 7. Diagrama da nuvem do Linking Open Data. (Fonte: [17])

O objetivo da arquitetura proposta é, como já mencionamos, gerar ontologias em OWL a partir dos mapas conceituais, tornando possível dispensar a presença de peritos em engenharia do conhecimento, durante todo o processo. Ou seja, queremos que a ferramenta resultante dessa arquitetura, o OntoMap, possa ser usada por especialistas de qualquer domínio, de maneira simples e amigável.

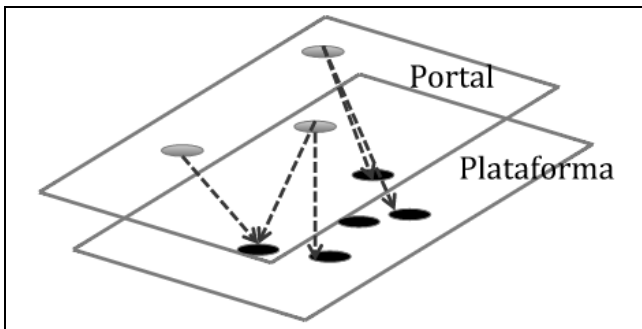


Figura 8. Visão macro da arquitetura SOA no projeto.

A Figura 9 apresenta a arquitetura do sistema OntoMap propriamente dito, destacando cada módulo do mesmo. Nela podemos observar os elementos que estão dispostos nas camadas de visão (Editor estendido e o módulo Criador) e os elementos da camada de serviços (os módulos Mapeador e Codificador), que serão explicados a seguir nas Subseções 4.1 e 4.2, respectivamente.

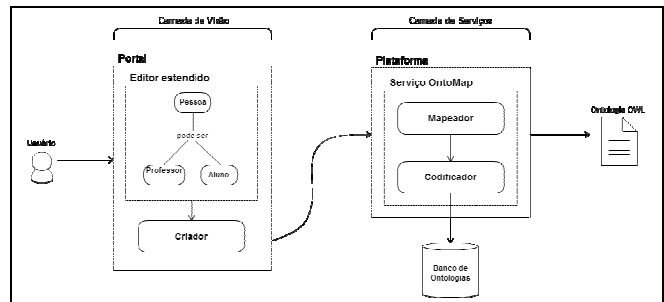


Figura 9. Arquitetura de sistema detalhada.

4.1 Elementos da Camada de Visão

A forma de acesso à ferramenta OntoMap se dá pelo uso do nosso Portal. O usuário acessa o editor de mapas conceituais normalmente, porém ao clicar em “OntoMap” que se encontra na aba lateral esquerda (Figura 10), o editor passa a fornecer algumas características adicionais para o usuário, transformando-se em algo que estamos chamando de Editor estendido. Importante salientar que, ao clicar nesse botão o usuário tem de estar ciente que o editor passa a ser adaptado para a construção de ontologias. A Seção 5 traz maiores informações sobre o Editor estendido do OntoMap.

Após a construção de um mapa, o módulo Criador é acionado. Ele fica encarregado de criar um arquivo em formato JSON correspondente ao mapa conceitual estendido criado pelo usuário. Além disso, o módulo Criador também é responsável pelo envio desse arquivo via *web*. Dessa forma, destacamos que é necessário que haja compatibilidade entre os arquivos gerados pelo módulo Criador e o arquivo de entrada recebido pelo módulo Mapeador (Subseção 4.2), na camada de serviço.

4.2 Elementos da Camada de Serviço

O módulo Mapeador contém uma lista de estereótipos de relações e seus respectivos *constructors* da linguagem OWL. Dessa forma, ao receber uma mensagem do módulo Criador, o módulo Mapeador se encarrega de realizar o mapeamento dessas relações atribuindo as axiomatizações necessárias. Nesse módulo também é realizado o mapeamento dos conceitos dos mapas conceituais em *Classes*, *Individuals* e *Datatype Properties* (elementos da linguagem OWL). Dessa forma, são criadas listas contendo os elementos respectivos para *Classes*, *Individuals*, *Datatype Properties* e relações, chamadas de *Object Properties* na OWL. Essas listas são passadas ao módulo Codificador.

Em posse das listas criadas pelo módulo Mapeador, o módulo Codificador só tem o trabalho de realizar de fato a conversão e codificar o mapa, a princípio em JSON, para o formato OWL. Além disso, o módulo Codificador se encarrega de salvar a ontologia OWL criada no Banco de Ontologias que, futuramente, será adaptado para se encaixar no padrão Linked Data [16] e posteriormente ligado ao DBpedia.

5. O PROTÓTIPO

Esta seção tem o objetivo de apresentar o atual estado de desenvolvimento do protótipo. Nele, estamos em fase de desenvolvimento do módulo Editor estendido do OntoMap, onde tomamos como base algumas das ferramentas apresentadas na Seção 3. Importante salientar também que o editor de mapas conceituais comum foi adaptado para que ele se tornasse apto a criar uma ontologia OWL. As decisões de projeto que tomamos para adaptar nosso editor foram listadas na Seção 3.1.

A Figura 10 ilustra o protótipo idealizado para a ferramenta OntoMap onde, ao criar uma nova relação, é exibida uma lista de estereótipos correspondentes com os *constructs* da linguagem OWL, que permitirá ao módulo Mapeador a construção de uma ontologia com maior fidelidade ao domínio.

Além da lista de estereótipos, o Editor estendido tem como característica a distinção visual dos principais elementos da OWL, como *Classes*, *Individuals* e *Datatype Properties*. Outra característica interessante que podemos destacar é que nosso Editor passa a ter abas que permitem ao usuário escolher se ele deseja exibir tais elementos no mapa, ou se deseja apenas exibir classes/conceitos (Figura 10). Dessa forma, combatemos uma das fraquezas listadas na Seção 3, que é a poluição visual dos mapas. Não consideramos adequado que um mapa misture conceitos com instâncias (*classes* com *individuals*) ou conceitos com atributos (*classes* com *datatype properties*). Assim, deixamos a critério do usuário se ele deseja ou não exibir determinados elementos da OWL no mapa.

Salientamos ainda que a Figura 10 mostra apenas uma versão *beta* de nosso protótipo que será atualizado ao longo do desenvolvimento de nosso projeto.

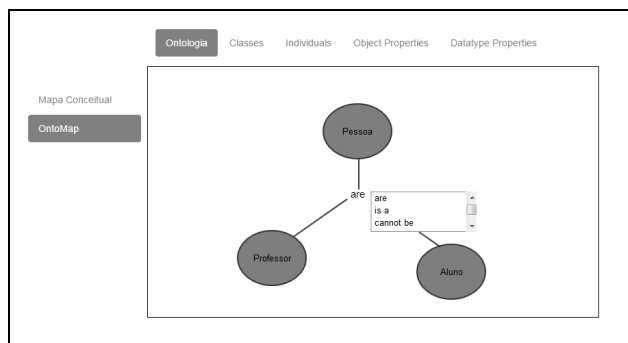


Figura 10. Protótipo do Editor OntoMap.

6. CONCLUSÃO E TRABALHOS FUTUROS

O tema Engenharia do Conhecimento está crescendo cada vez mais e dessa forma, novas formas de representação de conhecimento passam a ser foco de estudos de muitos pesquisadores. Tanto Mapas Conceituais quanto Ontologias são instrumentos de representação de conhecimento, cada um com seu propósito específico e nível de complexidade na construção de modelos. Por um lado, os mapas são informais e de fácil uso, já por outro, as ontologias são formais e complexas, o que demanda um especialista em engenharia de *software* para criá-las. Portanto, esse trabalho buscou unir esses dois mundos dando a possibilidade aos especialistas de domínio a criação de ontologias OWL, que é o padrão adotado pela Web Semântica, para então ser

possível que essa ontologia criada possa ser compartilhada na *web* para uso de toda a comunidade.

Cada uma das ferramentas analisadas na Seção 3 serviram de base para a idealização da ferramenta OntoMap. No entanto, o OntoMap difere-se das demais por fazer parte de um ambiente muito maior, ou seja, nossa solução faz parte de uma plataforma *web* que engloba uma série de aplicativos na qual nosso serviço será hospedado. Buscamos manter as principais vantagens dessas ferramentas analisadas e eliminar o que consideramos como debilidades. Para nosso editor, por exemplo, unimos as duas ferramentas no qual tivemos acesso para *download*, onde decidimos por manter os estereótipos do COE e as abas provenientes do MAP2OWL. Dessa forma, fomos capazes de manter a simplicidade do editor de mapas conceituais, sem poluir visualmente a visão do usuário, dando a possibilidade para o mesmo decidir o que exibir na tela.

A própria arquitetura de sistema na qual este trabalho foi baseado, a arquitetura SOA (*Service Oriented Architecture*), já é considerada outra inovação de nossa solução. Através dessa arquitetura, é possível que um sistema seja projetado em camadas, permitindo que a lógica de negócio seja desacoplada da camada de visão. Assim, este trabalho foi todo pensado baseando-se nesse modelo de arquitetura, onde a forma de comunicação entre as camadas se dá por meio do envio de mensagens. Como vantagens do SOA, podemos destacar que um serviço publicado na *web*, além de poder ser consumido por várias aplicações clientes, ele pode ser estendido e ampliado da forma que o cliente necessitar.

Importante dizer que este trabalho está apenas em uma versão inicial deste projeto, ou seja, uma versão *beta*. Dessa forma, nos preocupamos apenas na criação de um protótipo, que posteriormente idealizamos torná-lo uma ferramenta, para então podermos realizar estudos de caso e avaliar sua qualidade operacional em serviço. Então, a partir daí, será possível a disponibilização para uso de toda a comunidade.

Para efeito de uso e validação das nossas ferramentas, estamos concluindo a edição de um curso sobre o uso dos mapas conceituais em ambientes virtuais para professores da rede pública do nosso estado, que deverá começar nos próximos meses. Dessa forma, garantimos a divulgação e uso de nossas ferramentas bem como sua validação e melhoria.

O uso do OntoMap pela comunidade será de grande importância, pois será possível obter *feedbacks* sobre a ferramenta, o que nos permitirá uma possível correção de rota quanto a usabilidade da ferramenta. Assim, os trabalhos futuros convergem tanto na melhoria do editor de mapas (parte visual) quanto no serviço existente, no qual buscaremos a incorporação de novos estereótipos das relações criadas em um mapa conceitual, a fim de mapear mais *constructors* da linguagem OWL.

Com o intuito de disponibilizar na *web* as ontologias rasas criadas em nossa plataforma, outro trabalho futuro que gostaríamos de destacar, é que buscaremos incorporar nosso banco de ontologias no Linking Open Data, principalmente através da ligação do OntoMap com o DBpedia.

7. REFERÊNCIAS

- [1] Ausubel, D. P. (2000) The Acquisition and Retention of Knowledge: A Cognitive View. Kluwer Academic Publishers, Boston.
- [2] Ausubel, D. P., Novak, J. D., & Hanesian, H. (1968) Educational psychology: A cognitive view.

- [3] Brachman, R. & Levesque, H. (2004) Knowledge Representation and Reasoning, Morgan Kaufmann Publishers Inc.
- [4] Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Gómez, G., Arroyo, M. & Carvajal, R. 2004. CmapTools: A knowledge modeling and sharing environment. In: Concept Maps: Theory, Methodology, Technology, Proceedings of the First International Conference on Concept Mapping. Universidad Pública de Navarra: Pamplona, Spain. p. 125-133.
- [5] Chen, P. (1976) The entity-relationship model: Towards a unified view of data, ACM Transactions on Database Systems 1.
- [6] Cury, D. & Menezes, C. S. (2012). A Portal of Knowledge based on concept maps. In: Conference on concept maps, Valetta. Proceedings of the CMC2012, 2012. v. 2.
- [7] Cury, D., Perin, W. & Santos Jr P.S. (2014) CMPaaS – A platform of services for construction and handling of concept maps. In Proceedings of the 6th international conference on concept mapping, Santos, Brazil.
- [8] DBpedia. Disponível em: <<http://wiki.dbpedia.org/>>. Acesso em: 7 de Outubro de 2015.
- [9] Dutra, I. M., Fagundes, L. D. C., & Cañas, A. J. (2004). Uma proposta de uso dos mapas conceituais para um paradigma construtivista da formação de professores a distância.
- [10] Garcia, A., Norena, A., Betancourt, A., García, L., Sequeda, J. F. (2008). CMAPS supporting the development of OWL ontologies. Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany.
- [11] Gomez-Gauchia, H. & Diaz-Agudo, B. (2004). Two-layered approach to knowledge representation using conceptual maps and description logics. IHMC, Pamplona, Spain.
- [12] Guarino, N. (1998) Formal Ontology and Information Systems. In Proceedings of International Conference in Formal Ontology and Information Systems, pp 3-15.
- [13] Guizzardi, G. (2005) Ontological foundations for structural conceptual models, PhD Thesis, Enschede: CTIT, Centre for Telematics and Information Technology.
- [14] Hayes, P., Eskridge, T. C., Reichherzer, T., Saavedra, R., Mehrotra, M. & Bobrovnikoff, D. (2005) COE: Tools for Collaborative Ontology Development and Reuse. In: Knowledge Capture Conference (K-CAP).
- [15] Kendal, S. L., & Creen, M. (2007). An introduction to knowledge engineering (pp. 1-25). Springer London.
- [16] Linked Data. Disponível em: <<http://linkeddata.org/>>. Acesso em: 7 de Outubro de 2015.
- [17] LOD (Linking Open Data). Disponível em: <<http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>>. Acesso em: 7 de Outubro de 2015.
- [18] Novak, J. D. (1977). A Theory of Education. Ithaca, NY: Cornell University Press.
- [19] Novak, J. D. & Cañas, A. J. (2008), The Theory Underlying Concept Maps and How to Construct and Use Them, Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition.
- [20] Novak, J. D. & Gowin, D. B. (1984). Aprender a Aprender. 1. ed. Lisboa: Plátano Edições Técnicas.
- [21] OWL (Web Ontology Language). Disponível em: <<http://www.w3.org/2001/sw/wiki/OWL>>. Acesso em: 7 de Outubro de 2015.
- [22] Perin, W. A., Cury, D. & Menezes, C. S. (2014). NLP-Imap: Integrated solution based on question-answer model in natural language for an inference mechanism in concepts maps. In Proceedings of the 14th International Conference on Concept Mapping.
- [23] Simón, A., Ceccaroni, L., Rosete, A. (2007). Generation of OWL Ontologies from Concept Maps in Shallow Domains. In: Conference of the Spanish Association for Artificial Intelligence, 12., Salamanca. Anais eletrônicos... Disponível em: <http://link.springer.com/chapter/10.1007%2F978-3-540-75271-4_27#page-1>. Acesso em: 01 dez. 2014.
- [24] Starr, R. R. (2009). Uso de mapas conceituais como suporte à aquisição autônoma de conhecimento. Master Dissertation, Instituto Tecnológico da Aeronáutica (ITA), Sao Paulo, Brazil. Tavares, Romero. 2007. "Construindo mapas conceituais." Ciências & Cognição 12: 72-85.
- [25] Studer, R., Benjamins, R., & Fensel, D. (1998) Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1):161-197.
- [26] UML (Unified Modeling Language). Disponível em: <<http://www.uml.org/>>. Acesso em : 7 de Outubro de 2015.
- [27] Vassoler, G. A., Perin, W. A. & Cury, D.. (2014). MergeMaps – A computacional tool for merging of concept maps. In Proceedings of the 14th International Conference on Concept Mapping.
- [28] VOWL (Visual Notation for OWL Ontologies). Disponível em: <<http://vowl.visualdataweb.org/>>. Acesso em: 7 de Outubro de 2015.
- [29] Zouaq, A., Nkambou, R., & Frasson, C. (2007). Document Semantic Annotation for Intelligent Tutoring Systems: A Concept Mapping Approach. In FLAIRS Conference.