

Onto4AllEditor: um editor web gráfico para construção de ontologias por todos os tipos de usuários da informação

Fabício Martins Mendonça

Doutor; Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil;

fabicio.mendonca@ice.ufjf.br

Lucas Piazzzi Castro

Graduando; Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil;

lpiazzzi@ice.ufjf.br

Jairo Francisco Souza

Doutor; Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil;

jairo.souza@ice.ufjf.br

Maurício Barcellos Almeida

Doutor; Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil;

priv.mba@gmail.com

Eduardo Ribeiro Felipe

Doutor; Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina, MG, Brasil;

erfelipe@hotmail.com

Resumo: Ontologias são artefatos de representação do conhecimento utilizados no âmbito da Ciência da Informação, principalmente no campo da Organização do Conhecimento, bem como em áreas relacionadas. Mesmo que o tema já venha sendo disseminado há mais de duas décadas, a construção de ontologias ainda traz desafios para profissionais, pesquisadores e estudantes da Ciência da Informação. Tais desafios estão relacionados, principalmente, ao jargão lógico-filosófico necessário, a etapa formalização de conhecimento e a complexidade dos editores disponíveis, voltados para a computação. A presente pesquisa objetiva trazer uma alternativa a tais desafios. Apresenta-se aqui o *Onto4AllEditor*, um editor de ontologias gráfico, colaborativo e baseado na Web, que inclui recursos simples e intuitivos. Baseado na metodologia *OntoForInfoScience*, o editor foi planejado para facilitar o trabalho de usuários com níveis de experiência variados, objetivando popularizar a construção de ontologias. O artigo debate as limitações dos editores populares disponíveis e os compara ao que é oferecido pelo *Onto4AllEditor*. Conclui-se que o *Onto4AllEditor*, mesmo que já inclua recursos avançados, facilita a construção

de ontologias, em particular, para aqueles usuários que estão iniciando nesse tipo de atividade.

Palavras-chave: Editor de ontologia. Construção de ontologia. Representação do Conhecimento.

1 Introdução

A ontologia é um tema há muito estudado na Filosofia e representa um ramo da Metafísica preocupado com o que existe (BLACKBURN, 1996). Do ponto de vista teórico, a Ciência da Informação (CI), como Ciência Social Aplicada, se aproximou de teorias da Filosofia na perspectiva de estudos ontológicos. Do ponto de vista do artefato, somente a partir da década de 1990 – em trabalhos como o de Soergel (1997) e Vickery (1997) – as pesquisas evoluíram e as ontologias passaram a ser consideradas um tipo de sistema de organização do conhecimento. Junto a outros artefatos como, por exemplo, os tesauros, compõe assim um espectro de alternativas para fins de representação, adicionando ali uma instância capaz de expressar conhecimento formal em domínios diversos.

Representações formais, como aquelas proporcionadas pelas ontologias, cresceram em importância nos últimos anos em diferentes campos científicos que lidam intensivamente com dados. Este aumento se deve principalmente a sua possibilidade de representação inequívoca dos dados, com vantagens sobre tecnologias tradicionais como os bancos de dados, além da geração de inferências automáticas (ALMEIDA, 2020). A Ciência da Computação, por exemplo, tem usado ontologias como meio de integração entre dados heterogêneos, interoperabilidade, organização de dados na Web Semântica, e como artefato de modelagem conceitual para sistemas de informação (GUARINO, 1998; SMITH, 2004; GUIZZARDI, 2005; MENDONÇA; ZAIDAN, 2019). Na Linguística, ontologias facilitam o tratamento de dados dispersos, da complexidade e da incompletude de dados multilíngues, de questões terminológicas, além de auxiliar no desenvolvimento da área com estudos sobre semântica formal (SCHALLEY, 2019).

O presente artigo aborda a ontologia, no contexto da Ciência da Informação, como um artefato para organização e representação da informação em domínios especializados. Nesse contexto, foi criado um projeto de pesquisa, ainda em andamento, para disseminar o desenvolvimento de ontologias no âmbito da CI, bem como áreas afins. A partir da metodologia de construção de ontologias específica para a CI (MENDONÇA, 2015) – a *OntoForInfoScience* – desenvolveu-se o *Onto4AllEditor*, um editor de ontologias gráfico, baseado na Web e dotado de funcionalidades básicas para a criação de ontologias. O objetivo do presente artigo é apresentar o novo editor e os recursos de suporte aos problemas apontados em outros editores. O *Onto4AllEditor*¹ visa facilitar e popularizar a construção de ontologias, tornando acessível o desenvolvimento de ontologias por usuários sem experiência na área.

De fato, a construção de ontologias é um processo complexo e oneroso, de forma que, no âmbito da CI, se mantém os desafios para uma ampla disseminação e uso desses artefatos, os quais envolvem a necessidade de conhecimento sobre: a) o jargão técnico da área, que é multidisciplinar; b) fundamentos básicos da Lógica e princípios da Filosofia; c) noções usuais em modelagem e orientação à objetos, como classes abstratas e concretas, herança, disjunção e sobreposição, cardinalidade, dentre outras; d) formalismos e linguagens lógicas para ontologias e Web Semântica, dentre elas *Resource Description Framework* (RDF) e *Ontology Web Language* (OWL). Esses são exemplos de conteúdo nem sempre familiares a estudantes e profissionais de Ciência da Informação, cujos currículos, em geral, não privilegiam questões aplicadas.

Acrescenta-se o fato de que o uso de editores populares, como o *Protégé* (MUNSEN, 2015), é complexa, trabalhosa e em língua estrangeira. Alguns problemas comuns, apontados em entrevistas com usuários, são: a) falta de ambiente colaborativo para especialistas de domínio; b) falta de suporte para visualização de ontologias; c) funcionalidades pobres de depuração de erros; d) falta de suporte a pesquisa de termos para reuso; e) interface pouco amigável na navegação pela hierarquia de classes; dentre outros (VIGO, 2014). Para solucionar questões desse tipo, o *Onto4AllEditor* tem incorporado

funcionalidades como: a) interface gráfica, colaborativa e na web; b) associação entre etapas da metodologia e tarefas de editor; c) console de avisos e erros; d) reúso de ontologias de alto nível; para citar algumas delas.

Finalmente, cabe enfatizar que nossa pesquisa aborda algumas dentre as muitas dificuldades em se construírem ontologias dentro do campo da CI. Sob essa perspectiva, faz-se necessário observar os editores em si, pois, de fato, essas ferramentas têm sua parcela na complexidade inerente ao processo de desenvolvimento de ontologias. Ainda que a dificuldade para o uso de editores seja discutida, não é objetivo do artigo colocar todo o ônus nas ferramentas existentes, mas sistematizar o estado da arte sobre tal temática e desta forma apresentar os problemas verificados pela literatura já existente.

O restante do presente artigo está organizado como segue: a seção dois aponta problemas no processo de construção de ontologias, destaca editores de ontologia, seus usos e limitações; a seção três descreve o *Onto4AllEditor*, suas características básicas, funcionalidades e melhorias em termos de usabilidade; a seção quatro avalia outros editores de ontologia, a partir de um conjunto de boas práticas; por fim, a quinta seção oferece considerações finais e os trabalhos futuros desta pesquisa.

2 Editores de ontologias: erros, limitações e alternativas

A atual disponibilidade de metodologias de construção, há muito estabelecidas – como por exemplo, a *Methontology* (GÓMEZ-PEREZ et al, 1996), o Método 101 (NOY; McGUNINESS, 2001) e a NeOn (SUÁREZ-FIGUEROA, 2008) – bem como a existência de editores de ontologias, não tem representado facilidades em construir ontologias, nem mesmo resultados livres de erro.

A literatura aponta ‘erros comuns’ de desenvolvimento, mesmo que existam diferentes abordagens e denominações, por exemplo: erros e padrões comuns (RECTOR *et al.*, 2004); anomalias (POVEDA-VILLALÓN; SUÁREZ-FIGUEROA; PEREZ, 2010 e 2012); antipadrões (ROUSSEY; CORCHO; VILCHES-BLÁZQUEZ, 2009) (SALES; GUIZZARDI, 2015); definições mal formadas (CEUSTERS *et al.*, 2004), (MUNN; SMITH, 2008); dentre outros.

Encontram-se também propostas de classificação desses erros em conceituais, lógicos, de consistência, de completude e de concisão (GANGEMI *et al.*, 2006) (POVEDA-VILLALÓN; SUÁREZ-FIGUEROA; PEREZ, 2010). Diante das diferentes denominações, opta-se aqui por denominar todos e genericamente, por ‘erros ontológicos’, para fins de simplicidade. Listas extensas desses erros podem ser encontradas – como por exemplo em Poveda-Villallón, Suárez-Figueroa e Perez (2012); Sales e Guizzardi (2015); Mendonça e Soares (2017) –, mas um grupo de erros básicos foi selecionado para ilustrar o tipo de situação a que se faz referência (Quadro 1).

Quadro 1 - Erros comuns na construção de ontologias.

Descrição do Erro	Exemplos	Referências
1. Circularidade na definição de classes	Vocabulário <i>Friend of a Friend</i> (FOAF), classe “documento”: “A classe documento representa aquelas coisas as quais, em um sentido amplo, são concebidas como documentos .”	Rector <i>et al</i> (2004) Munn e Smith (2008) Poveda-Villalón, Suárez-Figueroa e Gómez-Perez (2010) Sales e Guizzardi (2015)
2. Sobrecarga de relações é_um (<i>is_a</i>), não especificando os diferentes tipos desta relação.	GlóbuloVermelho is_a Célula (gênero-espécie ou subsunção) GlóbuloVermelho is_a Eritrócito (equivalência ou sinonímia)	Guarino e Welty (2004)
3. Inclusão de relações todo-parte (<i>has_part</i> e <i>part_of</i>) ambíguas e inconsistentes, não especificando os diferentes tipos de relação todo-parte existentes.	SistemaReprodutivoFeminin o part_of SerHumano MoléculasDeCarboidrato part_of Intestino	Bittner e Donnelly (2007) Keet e Artale (2008) Schulz, Bittner e Kumar (2006)
4. Não preenchimento das relações inversas	Auto-explicativo	Poveda-Villalón, Suárez-Figueroa e Gómez-Perez (2010)
5. Não preenchimento das propriedades <i>domain</i> e <i>range</i> da relação	Auto-explicativo	Poveda-Villalón, Suárez-Figueroa e Gómez-Perez (2010)
6. Definição de relações	As relações é_vendido_em e	Poveda-Villalón, Suárez-Figueroa e

inversas incorretamente	é_comprado_em não são relações inversas	Gómez-Perez (2010)
7. Uso de heranças múltiplas na hierarquia de classes	VinhoDoPorto is_a VinhoVermelho VinhoDoPorto is_a VinhoSobremesa	Munn e Smith (2008) Noy e McGuinness (2001)
8. Inclusão de ciclos na hierarquia de classes (confusão uso-menção)	Professor como subclasse de Pessoa; Pessoa como subclasse de Professor.	Poveda-Villalón, Suárez-Figueroa e Gómez-Perez (2010) Sales e Guizzardi (2015)

Fonte: Elaborado pelos autores (2020).

Parte dos problemas e erros na construção de ontologias podem ser creditados a limitações dos editores, os quais exigem conhecimento prévio sobre as complexas teorias que envolvem Filosofia e Lógica. Tal complexidade impõe dificuldades, especialmente, para usuários iniciantes. Aborda-se aqui, esta e outras questões sobre editores de ontologias: lista-se os principais editores de ontologia com base em referências do W3C² e do grupo *Industrial Ontologies Foundry*³ (IOF) (subseção 2.1); descrevem-se pesquisas sobre uso e avaliação de editores por usuários (subseção 2.2); apontam-se limitações dos editores disponíveis a partir de testes realizados na presente pesquisa e de melhorias identificadas na literatura (Seção 3.3).

2.1 Sobre editores de ontologia conhecidos e populares

A presente subseção lista, de forma não exaustiva, editores citados em três referências atualizadas e relevantes da área (vide Quadro 2), a saber: W3C (2020); IOF (2020); BRAUN, ESTEVEZ e FILLOTTRANI (2018). Os editores estão listados em ordem alfabética e separados em **editores disponíveis** e **editores inativos**.

Quadro 2 – Exemplos de editores de ontologia.

Editores Disponíveis

Editor	Características	Referência
CmapTools	Editor de mapeamento conceitual que permite representar nós e relações em ambiente gráfico. Para utilização colaborativa conta com o servidor CmapServer e possui versão para utilização em <i>tablets</i> . A licença disponível inclui livre distribuição, pessoal ou comercial.	https://cmap.ihmc.us/cmaptools/
Eddy for Graphol	Editor gratuito que usa a linguagem gráfica Graphol. A ontologia é desenvolvida através de grafos compostos apenas por nodos e conexões, evitando a construção textual e sintaxes de axiomas. A linguagem gráfica permite a definição de predicados, constructos e conexões de inclusão e entrada, dentre outros.	https://www.obdasystems.com/eddy
Hozo	Editor em um ambiente distribuído que possui arquitetura em três componentes: editor, servidor e o gerenciador de ontologia. O editor possui interface gráfica, permite construir ontologia em ambiente Web, e evita a exposição código OWL ao usar gráficos.	http://www.hozo.jp/
Menthor	Editor gratuito, de código aberto, que possui integração com a OntoUML na simulação visual do processo. Conta com estereótipos de diagramas da <i>Unified Model Language</i> (UML) em OntoUML, além de permitir importação do <i>Enterprise Architect</i> via <i>plug-in</i> .	https://github.com/MenthorTools/menthor-editor
OntoUML Lightweight Editor (OLED)	É um editor para implementação de ontologias em OntoUML, ligada a metodologia da <i>Unified Foundational Ontology</i> (UFO). Oferece validação sintática, simulação visual, validação de modelo e referências, detectando padrões ontológicos e antisseânticos. Também suporta integração com o <i>Sparx System Enterprise Architect</i> para importação de modelos.	https://github.com/nemo-ufes/ontouml-lightweight-editor
Onto4AllEditor	Editor com interface gráfica e colaborativa na web e baseado na metodologia <i>OntoForInfoScience</i> . Objetiva popularizar a construção de ontologias, facilitando o uso por usuários sem experiência. Além de interface gráfica, inclui: associação entre metodologia e tarefas do editor; console	https://onto4alleditor.com/

	de avisos; reuso de ontologias; relatório de erros; e editor de tesauros.	
OWLGrEd	Editor gratuito com interface gráfica para edição e visualização de ontologias. A principal característica é a visualização gráfica de grandes ontologias, mas possui notação para operações sobre entidades, definições, propriedades de objetos e restrições. A extensibilidade do editor é feita por <i>plug-ins</i> , incluindo modelagem em UML, por exemplo.	http://owlgred.lumii.lv/
Protégé	Editor gratuito, de código aberto, o mais usado pela comunidade acadêmica, governamental e corporativa. A versão desktop foi desenvolvida em arquitetura multiplataforma permitindo seu uso em sistemas operacionais diversos via <i>Java Virtual Machine</i> (JVM). Permite a extensão de funcionalidades através de <i>plug-ins</i> , além de dispor de um processador de consultas em lógica descritiva e trabalhar com as diversas sintaxes do W3C (OWL, RDF, <i>Turtle</i>).	https://protege.stanford.edu/
TopBraid Composer	Ambiente de desenvolvimento integrado de modelagem semântica com múltiplas funções, trata-se de um software comercial da TopQuadrant, com recursos que permitem a criação de modelos ontológicos e <i>linked data</i> em interface gráfica. Importa dados em diversos formatos (XML, XMLS, Planilhas, Bancos de Dados, JSON, dentre outros), tem capacidades de inferência, mapeamento, geração de consultas em linguagem SPARQL e apoio de sintaxe automática.	https://www.topquadrant.com/products/topbraid-composer/
yEd Graph Editor	É um editor para desktop para a criação de diversos tipos de diagramas, que possui capacidade nativa para grafos de redes semânticas e aceita como extensão a especificação Graffo para apresentação em OWL.	https://www.yworks.com/products/yed
WebProtégé	WebProtégé é uma versão do <i>Protégé</i> na Web com funcionalidades reduzidas. Suporta edição em ontologias na OWL 2.0 e <i>Open Biomedical Ontologies</i> (OBO); possibilita importar e exportar ontologias em formatos como: RDF/XML, <i>Turtle</i> , OWL/XML, OBO, dentre outros. O principal atrativo é colaboração simultânea, via Web, para	https://webprotege.stanford.edu/

	diversos atores.	
Editores Inativos		
Editor	Características	Referência
SWOOP	Editor da <i>University of Maryland</i> , inspirado na simplicidade da interface dos navegadores web, com elementos de interação gráfica simplificados. Sua última release data de janeiro de 2006, na versão 2.3. Possuía suporte a <i>plug-ins</i> e <i>reasoners</i> , como Pellet, além de formatos RDF/XML.	https://github.com/ronwalf/swoop
NeOn Toolkit	Desenvolvido em um projeto iniciado em 2006 pela <i>European Commission's Sixth Framework Programme</i> , com duração de 4 anos. O <i>NeOn</i> aborda diversos artefatos para o desenvolvimento de ontologias em um ambiente distribuído, contando com um editor próprio. Foi descontinuado com uma última versão em 2011.	http://neon-project.org/
OBO-Edit	Editor de código aberto, especializado para o formato biomédico OBO, tem como características possuir uma interface fácil para edição, um raciocinador automático próprio e artefatos de navegação e pesquisa.	http://oboedit.org/

Fonte: Elaborado pelos autores (2020).

2.2 Sobre a utilização de editores de ontologia

Para verificar como os editores de ontologia são vistos e utilizados, realizou-se uma revisão de literatura não sistemática, a qual retornou duas dezenas de trabalhos nos últimos dez anos. Essas pesquisas trazem impressões sobre funcionalidades e dificuldades em usar os editores.

Uma pesquisa quantitativa com 65 profissionais que trabalham com ontologias identificou, dentre outras questões, os editores preferidos (WARREN, 2013). O *Protégé* apareceu como o mais utilizado, na preferência de 50% dos entrevistados. O *TopBraid Composer* apareceu em segundo lugar com 14%. O *NeOn Toolkit* (MICHAEL; WATERFELD, 2012), o *CmapTools* (CAÑAS *et al*, 2004) e o *SWOOP* (KALYANPUR *et al*, 2006) apareceram, respectivamente, com 6%, 5% e 4%. Por fim, um editor para Biomedicina, o

OBO Edit (DAY-RICHTER; HARRIS; HAENDEL, 2007) foi citado por 3% dos entrevistados.

Outra pesquisa apresentou aos participantes, tarefas em cinco editores distintos, a saber: *TopBraid Composer*, *Protégé*, *SWOOP*, *NeOn Toolkit* e *CmapTools* (MALIK, 2017). Em seguida foi aplicado um questionário de usabilidade aos participantes. Dentre os resultados, o *Protégé* apareceu como o mais fácil de usar e aquele que exigiu menos tempo para as tarefas, ao passo que o *IHMC CmapTool* apresentou os piores resultados. O tempo para as tarefas com o *Protégé* foi cerca da metade em comparação com outros editores como *TopBraid*, *NeOn* e *SWOOP*. No teste de usabilidade, *TopBraid Composer* conseguiu o melhor resultado, seguido pelo *Protégé*, *SWOOP*, *NeOn Toolkit* e *CmapTools*, respectivamente. Apesar de ser o melhor pontuado em usabilidade, o *TopBraid* não foi considerado o melhor em critérios como curva de aprendizado do software, tempo para execução das tarefas e disponibilidade. Considerando a classificação geral, feita pelos usuários da pesquisa, da melhor ferramenta, tem-se na sequência: o *Protégé*, seguido do *TopBraid Composer*, do *NeOn Toolkit*, do *SWOOP* e do *IHMC Cmap Tools*.

Em entrevista com cerca de 30 profissionais de ontologias, questionou-se qual o melhor editor para iniciantes (SIRICHAROEN, 2018). O *Protégé* foi o indicado como o mais adequado e, dentre os outros, apenas o *TopBraid Composer* teve mais de uma menção. Os resultados mostraram que cerca de 70% dos profissionais recomendam o *Protégé* e apenas 10% indicaram o *TopBraid Composer*. As demais ferramentas envolvidas foram *NeOn Toolkit*, *SWOOP*, *OntoStudio*. Um dos aspectos mencionados por aqueles que escolheram o *Protégé* foi o tutorial (pizza). Outros motivos que incluem sua grande comunidade de usuários, é o fato de ser gratuito e de ter código aberto.

2.3 Limitações atuais

Até aqui, foi apresentada uma lista de editores de ontologia, além de trabalhos sobre a sua utilização e de avaliação pelos usuários. Com base nessas referências, foram compiladas as limitações dos editores que influenciam, de

alguma maneira, a atuação de usuários que desejam construir ontologias. Foram identificadas como principais limitações:

- a) Problemas com a visualização gráfica, seja a ausência dela ou a sua navegação pouco amigável: poucas ferramentas permitem que o usuário interaja dinamicamente com o modelo gráfico da ontologia, especificando classes, relações e propriedades;
- b) Falta de suporte para visualização de ontologias nos ambientes selecionados e preferidos pelos usuários, assim como interface pouco amigável na navegação pela hierarquia de classes (VIGO, 2014);
- c) Necessidade de um ambiente colaborativo entre os especialistas do domínio, pois um fator prejudicial nesse sentido é o fato de que muitos editores de ontologia não estão disponíveis na web (ALATRISH, 2013; VIGO, 2014);
- d) Falta de suporte metodológico implícito no desenvolvimento da ontologia do editor, como, por exemplo, ações que orientem os usuários nas tarefas (ALATRISH, 2013);
- e) O tempo dispendido para executar tarefas de construção da ontologia em editores é grande devido a sua complexidade (MALIK, 2017; SIRICHAROEN, 2018);
- f) Resultados pobres na depuração de erros da ontologia (VIGO, 2014);
- g) Falta de suporte para pesquisa de termos em ontologias externas para fins de reuso (ALATRISH, 2013; VIGO, 2014);
- h) Grande variedade de sintaxes de linguagens formais para importação e exportação (ALATRISH, 2013), o que pode prejudicar a interoperabilidade semântica;
- i) Falta de suporte aos resultados e ao funcionamento de *reasoners* em alguns editores (ALATRISH, 2013; VIGO, 2014);
- j) A extensibilidade dos editores é assegurada, em grande parte, pela inclusão de *plug-ins*, o que aumenta a sua complexidade (ALATRISH, 2013);
- k) Muitos editores de ontologias foram descontinuados, não estão mais disponíveis ou já não recebem mais atualizações;

- l) Editores de ontologia comerciais, em geral, tem licença de alto preço, o que sugere que podem ser usados apenas por empresas;
- m) Falta de suporte à inserção, resultado e falhas em axiomas (VIGO, 2014);
- n) Não há um padrão para armazenamento nos editores, que podem estar ali em arquivos HTML e XML ou em bancos de dados relacionais (ALATRISH, 2013);
- o) Falta de suporte e gestão de *backup* das ontologias, em particular, aos baseados na Web (ALATRISH, 2013).

Tais limitações sugerem que os editores de ontologia disponíveis, de maneira geral, ainda necessitam de melhorias do ponto de vista de funcionalidades e facilidade de uso por seus usuários. Essa é uma das razões do projeto ao qual pertence a presente pesquisa, além das já conhecidas dificuldades de estudantes, pesquisadores e profissionais de Ciência da Informação (CI) na construção de ontologias. Cabe destacar também que o *Onto4AllEditor* não é solução para todas limitações e erros ontológicos, porém contribui sobremaneira para o uso de ontologias em Ciência da Informação (CI) por ter sido concebido e construído por profissionais da área, que conhecem as dificuldades e planejaram funcionalidades amigáveis e instruções para as tarefas de forma acessível.

3 Onto4AllEditor: características, funcionalidades e etapas

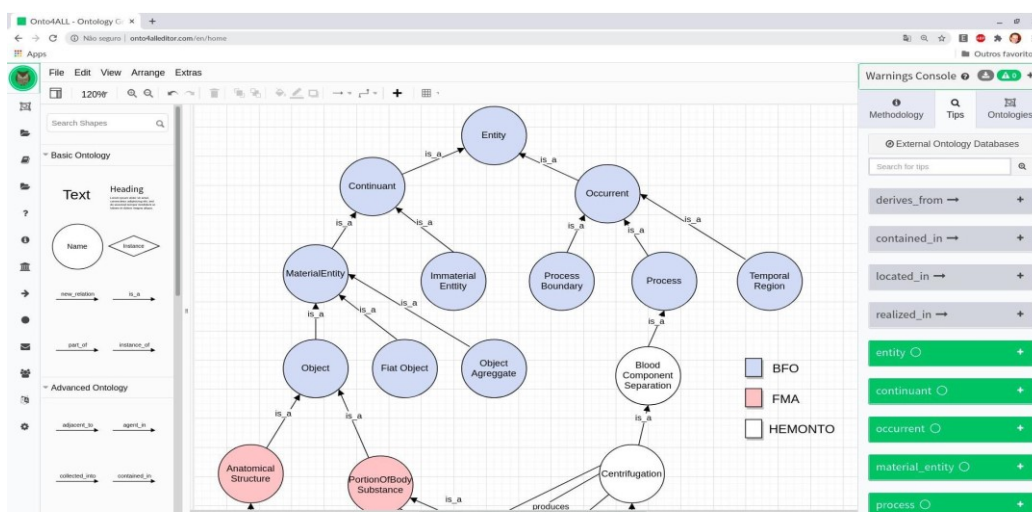
A presente seção descreve o *Onto4AllEditor* apresentando características, funcionalidades e etapas para criação de ontologias no editor. Inicialmente, é importante mencionar que a teoria que fundamenta o *Onto4AllEditor* é a metodologia *OntoForInfoScience* (MENDONÇA, 2015). Mendonça (2015) explica que a metodologia tem como característica principal uma abordagem detalhada das etapas e tarefas do ciclo de desenvolvimento ontológico, de forma a possibilitar melhor compreensão das atividades necessárias à construção de ontologias para diferentes tipos de desenvolvedores, dos menos aos mais

experientes. Esse também é um dos propósitos do *Onto4AllEditor*. Embora o editor se oriente pela *OntoForInfoScience*, trata-se de software independente e, por isso, pode ser utilizado com qualquer outra metodologia ou mesmo sem seguir alguma. As etapas da *OntoForInfoScience*, no escopo editor, servem para orientar os usuários nas etapas necessárias de construção, mesmo que segui-las não seja obrigatório.

O diferencial do *Onto4AllEditor* em relação aos demais editores avaliados corresponde às facilidades oferecidas no desenvolvimento por usuários sem experiência. Nesse sentido, foram incorporadas as seguintes funcionalidades principais: a) interface gráfica e colaborativa, disponível na web; b) suporte metodológico para tarefas executadas no editor; c) reúso de ontologias de alto nível; d) console de avisos e erros dinâmico (em tempo real) do modelo gráfico da ontologia; e) relatórios de erros e do conteúdo ontológico; f) editor de tesouros; dentre outros recursos.

A tela principal do editor é intuitiva e inclui uma área de desenho gráfico para modelagem da ontologia no ambiente Web (Figura 1). As funcionalidades do *Onto4AllEditor* são acessadas através de menu dinâmico, à esquerda da tela, com um ícone específico para cada possibilidade. Também à esquerda da interface está a paleta de elementos gráficos (classes e relações), a qual foi separada em dois grupos: a) *Basic Ontology*: contém apenas classes, e relações básicas *is_a*, *part_of*, *instance_of*, além da *new_relation* (para relações do usuário); b) *Advanced Ontology*: contém outras relações ontológicas – por exemplo, *derives_from*, *contained_in*, *located_in*, *realized_in* – pré-cadastradas no editor e oriundas de ontologias de alto nível - *Basic Formal Ontology* (BFO) e a *Relation Ontology* (RO).

Figura 1 - Tela principal do *Onto4AllEditor* e sua interface gráfica.



Fonte: Elaborado pelos autores.

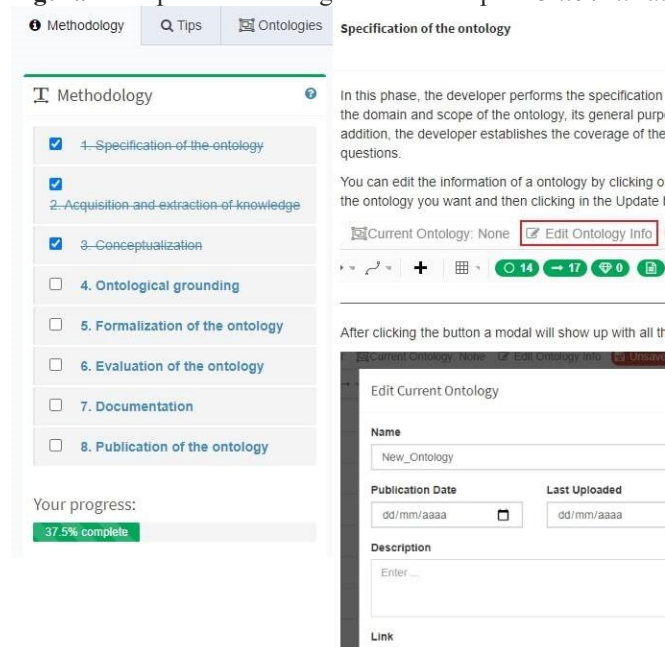
Cores e legendas podem ser usadas para informações relevantes como, por exemplo, classes importadas. Na Figura 1, apresenta-se um fragmento da HEMONTO, uma ontologia sobre hemoterapia (MENDONÇA; ALMEIDA, 2013) que importa classes da BFO e do *Foundational Model Anatomy* (FMA). O reúso de ontologias é uma prática recomendada na literatura da área (DEGEN *et al.*, 2001) (GRENON; SMITH, 2004) que está presente tanto na *OntoForInfoScience* quanto no *Onto4AllEditor*. O editor fornece a funcionalidade de busca às classes e relações cadastradas para facilitar o reúso, conforme apresentado na paleta à esquerda (Figura 1), na qual classes são representadas em cor mais escura e relações em cinza mais claro.

Sobre o suporte metodológico, o *Onto4AllEditor* é capaz de realizar sete etapas das oito previstas na *OntoForInfoScience*, a saber: 1) Especificação; 3) Conceitualização; 4) Fundamentação; 5) Formalização; 6) Avaliação; 7) Documentação; 8) Publicação. A segunda etapa da metodologia - aquisição e

extração de conhecimento - é a única não realizada, pois envolve aquisição do conhecimento do domínio a partir de fontes de informação.

De forma a orientar o desenvolvedor, as etapas da metodologia são exibidas em uma caixa de diálogo com progressão (vide Figura 2). Ao acessar uma etapa específica, o editor abre outra caixa de diálogo (vide Figura 2) contendo a descrição da etapa e as tarefas a executar. A Figura 2 explica a primeira etapa – especificação da ontologia – e como realizá-la a partir de um *template*.

Figura 2 - Suporte metodológico fornecido pelo *Onto4AllEditor*.



Fonte: Elaborado pelos autores (2020).

As etapas três, quatro e cinco são realizadas via comandos disponíveis na interface, com funções que possibilitam a inserção de elementos obrigatórios, que incluem:

- Inserção de classes, relações e instâncias (paleta esquerda, Figura 1)
- Definição de propriedades de classes e de relações (ao clicar em uma classe ou relação é aberta uma caixa de diálogo com suas propriedades, vide Figura 3). A Figura 3 ilustra o preenchimento das propriedades de uma classe de uma dada ontologia. No exemplo, estão sendo definidas as propriedades da classe “*portion of body substance*”.

- c) Inserção de axiomas com validação da sintaxe (mesma caixa de diálogo de propriedades através dos campos *SubClassOf*, *DisjointWith* e *EquivalentTo*, vide Figura 3);
- d) Especificação de questões de competência (*template* da etapa especificação).
- e)

Figura 3 - Caixa de diálogo para preenchimento das propriedades de classes e relações da ontologia.

ID:	27	x
importedFrom:	EMA	x
hasOBONamespace:		x
hasURI:	http://purl.org/sig/ont/fma/fma9669	x
hasSynonym:	Body substance	x
hasExactSynonym:		x
hasRelatedSynonym:		x
alternativeTerm:		x
definition:	Material anatomical entity in a	x
elucidaton:		x
exampleOfUsage:	portion of saliva, portion of blood,	x
SubClassOf:	is_a 'object aggregate' and ('has	x
comments:	produced by anatomical structures or	x
DisjointWith:		x
EquivalentTo:		x

Enter Property Name Add Property

☐ Placeholders Cancel Apply

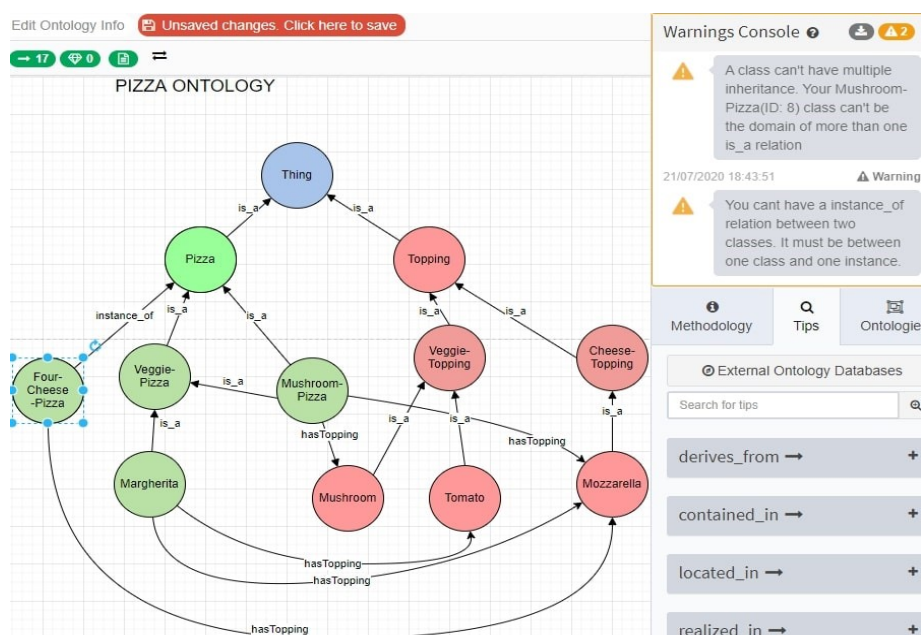
Fonte: Elaborado pelos autores (2020).

A sexta etapa, de avaliação da ontologia, foi implementada através de um console de avisos (vide Figura 4). Ao contrário dos erros detectados em programação, o *Onto4AllEditor* apenas alerta seus usuários sobre erros metodológicos, os quais não são de correção obrigatória. Os avisos do editor são mostrados em tempo real ao usuário em uma caixa de diálogo (canto superior direito da tela, como mostra a Figura 4) e podem ser exportados para arquivo-texto.

Os avisos até então implementados no *Onto4AllEditor* abrangem os seguintes itens: a) classes duplicadas; b) uso incorreto da relação “*instance_of*”; c) herança múltipla; d) circularidade; e) uso indevido de relações inversas; f) falta de anotações e metadados; g) erros no uso do ambiente gráfico.

A Figura 4 ilustra avisos no console para uma ontologia de testes, sobre pizzas, onde o usuário recebe dois avisos (em destaque): a) herança múltipla da classe *Mushroom-Pizza*, que é subclasse de *Pizza* e de *Veggie-Pizza*; b) uso incorreto da relação “*instance_of*” entre as classes *Four-Cheese-Pizza* e *Pizza*, já que tal relação conecta instância à classe. Em ambos os casos, o console detecta dinamicamente os problemas, informando o horário, a classe ou relação, além de emitir um identificador do aviso.

Figura 4 - Console de avisos de ontologia construída no *Onto4AllEditor*.



Fonte: Elaborado pelos autores (2020).

A etapa sete, de documentação, é realizada no *Onto4AllEditor* ao longo de todo o processo. Além disso, o usuário pode acessar um repositório específico para gerenciar suas próprias ontologias e exportar o conteúdo da ontologia para um arquivo-texto (vide Figura 5).

Figura 5 - Relatório de ontologia desenvolvida no *Onto4AllEditor*.

```

/***** Ontology Report *****/
Classes:
  Entity
  -----
  Object
  -----
  AnatomicalStructure
  -----
  Erythrocyte
  -----
  Process
  -----
  Centrifugation
  -----
  PortionOfBlood
  -----
Relations:
  AnatomicalStructure is_a Object
  -----
  Erythrocyte is_a Cell
  -----
  Process is_a Occurrent
  -----
  Centrifugation is_a BloodComponentSeparation
  -----
  PortionOfBlood participates_in Centrifugation
Properties:
  - label: participates_in
  - domain: Object
  - range: Process
  - inverseOf: has_participant
  - cardinality: 1
  - id: 26
  -----
  
```

Fonte: Elaborado pelos autores (2020).

A oitava e última etapa, de disponibilização da ontologia, envolve a exportação para um dos três formatos disponíveis no editor: imagem (.svg), texto semi-estruturado (.xml) e lógica (.owl). A exportação para lógica permite que a ontologia possa ser manuseada em outros editores, por exemplo, no popular *Protégé*, caso necessário.

4 Avaliação dos editores de ontologia

A partir da revisão dos principais editores de ontologia disponíveis (Seção 2) e da apresentação da ferramenta *Onto4AllEditor* (Seção 3), torna-se possível conduzir uma comparação de tais editores, traçando um panorama das características e funcionalidades de cada ferramenta, bem como de suas limitações e dificuldades dos usuários no âmbito da CI. O objetivo da presente seção é apresentar a avaliação dos editores e destacar as melhorias para construção de ontologias a partir do *Onto4AllEditor*.

Primeiramente, apresenta-se a avaliação dos editores já mencionados neste trabalho. Foi estabelecido um conjunto de critérios para a análise, a partir revisão de literatura sobre o tema (ALATRISH, 2013; VIGO *et al.*, 2014; WARREN, 2013; MALIK, 2017; SIRICHAROEN, 2018). Além disso, novos

critérios detectados na prática da construção de ontologias foram adicionados, tais como:

- a) O editor permite visão gráfica da ontologia?
- b) O editor está disponível on-line na web?
- c) O editor fornece suporte ao trabalho colaborativo na web?
- d) O editor oferece suporte metodológico às etapas de construção?
- e) Permite importar ontologias em quais linguagens formais?
- f) Permite exportar ontologias em quais linguagens formais?
- g) Fornece suporte à inserção de axiomas?
- h) Inclui *reasoners*?
- i) Oferece extensibilidade através por *plug-ins*?
- j) Está disponível em que tipo de ambiente (desktop, web ou outro)?
- k) Qual a data da última atualização do editor?
- l) Como são armazenadas as ontologias no editor?
- m) Fornece suporte e gerenciamento de backup das ontologias construídas?
- n) Para quais plataformas e sistemas operacionais está disponível?
- o) Qual o tipo de licença do software?
- p) O editor encontra-se disponível ou foi descontinuado?
- q) É um editor de ontologias nativo?

De acordo com tais critérios sugeridos para avaliação dos editores de ontologia revisados por esta pesquisa, construiu-se o Quadro 3 (a seguir), que apresenta o resultado da avaliação dos editores de ontologias aqui realizado.

Quadro 3 - Avaliação dos editores de ontologia.

	CmpTools	Eddy for Graphol	Hozo	ICOM	Menthor	NeOn Toolkit	OBO-Edit	OLED	Onto 4All Editor	OWLGr Ed	Protégé	SWOOP	TopBraid Composer	Web Protégé	yEd Graph Editor
1.Visão gráfica da ontologia	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
2.Disponível em ambiente web	Desktop + servidor	Não	Não	Não	Não	Não	Não	Não	Sim	Não	Não	Não	Não	Sim	Sim
3.Suporte ao trabalho colaborativo	Sim	Não	Sim	Não	Não	Não	Não	Não	Sim	Não	Não	Não	Sim	Sim	Sim
4.Suporte metodológico	Não	Não	Não	Não	Não	Não	Não	Não	Sim	Não	Não	Não	Não	Não	Não
5.Importação da ontologia em linguagens formais	CXL, XTM / XCM, IVML	OWL, graphml, graphol	ONT	UML XMI	XMI, EA	OWL	OBO, OWL (<i>plug-in</i>)	OLED, XML, XMI	OWL, XML	OWL	OWL, RDF, OBO	RDF, OWL	OWL, RDF, XML	OWL, RDF	graphml (<i>plug-in</i> da Graffo)
6. Exportação da ontologia em linguagens formais	CXL, XTM / XCM, IVML, PDF, os, SVG	OWL, XML, PDF	Text/HTML, XML-DTD, DAML+OIL, RDF, OWL	XML, DIG, UML XMI	XMI, UML, ECORE	OWL, RDF, OMN, TTL	OBO, OWL (<i>plug-in</i>)	XMI, UML	OWL, XML	OWL	OWL, RDF, OBO	RDF, OWL, HTML	OWL, RDF, XML	RDF/X ML, OWL	graphml (<i>plug-in</i> da Graffo)
7.Suporte a inserção de axiomas	Não	Sim	Não	Sim	Não	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim	Sim	Não
8.Suporte a reasoner	Não	Sim	Não	Sim	Não	Sim	Não	Sim	Não	Não	Sim	Sim	Sim	Não	Não
9.Extensibilidade (suporte a	Não	Sim	Não	Não	Não	Sim	Sim	Não	Não	Sim	Sim	Sim	Não	Não	Sim

<i>plug-ins</i>)															
10. Disponibilidade	Desktop, Server e iPad	Desktop	Desktop	Desktop	Desktop	Desktop	Desktop	Desktop	Web	Desktop	Desktop	Desktop	Desktop e Web	Web	Desktop e Web
11. Última atualização	Outubro, 2019	Março, 2020	Abril, 2019	Agosto, 2010	Fevereiro, 2019	Dezembro, 2011	Março, 2014	Setembro, 2015	Junho, 2020	Setembro, 2019	Março, 2019	Março, 2007	Agosto, 2018	Agosto, 2019	2020
12. Armazenamento das ontologias	Arq. Local ou Arq. no Servidor	Arquivo local	Arquivo local	Arquivo local	Arquivo local	Arquivo local	Arquivo local	Arquivo local	Arquivo local e Banco de dados	Arquivo local	Arquivo local	Arquivo local	Arquivo local ou Banco de dados	Servidor (opção de download)	Arquivo local ou Arq. no Servidor
13. Suporte e gerenciamento de backup	Não	Não	Não	Não	Não	Não	Não	Não	Através do banco de dados	Não	Não	Não	Não	Não	Não
14. Sistema operacional	Windows, macOS e Linux	Windows, macOS e Linux	Windows e macOS	Windows, macOS e Linux	Windows, macOS e Linux	Windows, macOS e Linux	Windows, macOS e Linux	Windows, macOS e Linux	Windows, macOS e Linux	Windows	Windows, macOS e Linux	Windows, macOS e Linux	Windows, macOS e Linux	Windows, macOS e Linux	Windows, macOS e Linux
15. Licença	Copyright	GNU v3	Copyright	Copyright	MIT	Creative Commons	Artistic License	Não informado	Copyright	Não informado	BSD 2-clause	MIT	Copyright	BSD 2-clause	Copyright
16. Status	Disponível	Disponível	Disponível	Descontinuado	Disponível	Descontinuado	Descontinuado	Disponível	Disponível	Ativo	Ativo	Inativo	Ativo	Ativo	Ativo
17. Editor de Ontologia nativo	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim	Não

Fonte: elaborado pelos autores (2020).

Uma análise geral do Quadro 3 permite identificar questões relevantes sobre os editores analisados. Em primeiro lugar, a maioria permite que o usuário interaja dinamicamente com o modelo gráfico da ontologia. Essa característica beneficia usuários com menos experiência e contribui para melhor visualização do modelo ontológico. Entretanto, a maioria dos editores não oferece a visualização do modelo da ontologia no ambiente e ferramenta preferidos do usuário, isto é, em outro software ou *plug-in*, mas apenas no próprio software onde a ontologia foi desenvolvida.

No que se refere à disponibilização do editor na web, nota-se que poucas ferramentas oferecem este recurso. Essa é uma característica importante para construção de ontologias por especialistas no domínio, pelo fato de possibilitar o trabalho colaborativo simultâneo e *on-line*. Esses recursos são nativos do *Onto4AllEditor*.

O suporte metodológico é outra funcionalidade de relevância para usuários menos experientes. Editores com essa característica, rara entre os avaliados, conseguem orientar seus usuários nas tarefas necessárias. Como citado antes, o editor *Onto4AllEditor* fornece suporte metodológico com base na *OntoForInfoScience*.

Com relação às funções de importação e exportação, existe uma grande variedade de sintaxes para uma linguagem formal – por exemplo, OWL/XML, RDF/XML, OWL 2.0, Turtle, Manchester Syntax etc. – e nem todos os editores avaliados fornecem suporte a esses diferentes tipos de linguagem formal. O *Onto4AllEditor*, nesse momento, permite importação e exportação em OWL/XML e está em processo de integração com a OWL-API para suportar todos os tipos de formatos recomendados pela W3C para ontologias – funcionalidade que será incluída na próxima versão do editor.

Outros recursos importantes são o suporte à inserção de axiomas, como como faz o *Protégé* através de um validador de expressões lógicas em Manchester Syntax, e a inclusão de *reasoners* para identificação de erros e inconsistências. O *Onto4AllEditor* permite, em sua versão atual, a inserção de axiomas, mas ainda não possui validador de expressões lógicas. A identificação de erros e inconsistências na ferramenta é realizada por abordagem diferente dos

reasoners, a partir do console dinâmico de avisos e erros, que fornece um feedback claro e interativo dos erros cometidos. A inclusão de *reasoners* como *plug-ins* do *Onto4AllEditor* está prevista para as próximas versões da ferramenta, já que eles são abordagens complementares ao console de erros na avaliação do conteúdo ontológico.

Sobre a disponibilização dos editores por ambiente e sistema operacional, não existem problemas na maior parte deles. Em geral, as ferramentas podem ser executadas nos sistemas operacionais Windows, Linux e MacOS. O *Onto4AllEditor* inclui-se nessa lista de softwares editores que podem ser executados nesses três sistemas operacionais. O ambiente mais comum de execução é o desktop e não o ambiente web. A disponibilização no ambiente Web ajuda a tornar colaborativo o trabalho, funcionalidade que o *Onto4AllEditor* já tem implementada.

Quanto ao armazenamento de ontologias, a maioria dos editores permite salvá-las em um arquivo local na máquina do usuário. O *Onto4AllEditor* dispõe desse tipo de armazenamento, mas também pode salvar o trabalho em um banco de dados no servidor web, recurso que permite o gerenciamento de ontologias desenvolvidas por cada usuário. Com relação ao gerenciamento e suporte de backup das ontologias de cada usuário, os editores avaliados que geram apenas arquivos locais não incluem esse recurso, diferentemente daqueles que utilizam o armazenamento em banco de dados, tal como o *Onto4AllEditor*.

Por fim, cabe destacar outras duas funcionalidades que são diferenciais do *Onto4AllEditor*, mas que não foram incluídas como critério de análise (Quadro 3): a) a documentação da ontologia, desenvolvida no editor através da geração de relatórios do conteúdo ontológico; b) o reúso de ontologias, com suporte à pesquisa de termos em ontologias e repositórios ontológicos externos.

5 Considerações finais

A presente pesquisa apresentou o editor web gráfico de ontologias, *Onto4AllEditor*, como alternativa para modelagem ontológica, principalmente, para não especialistas em ontologias. O editor se fundamenta na

OntoForInfoScience, uma metodologia de atividades do ciclo de desenvolvimento ontológico para a CI. A metodologia e o editor fazem parte de um projeto em andamento que objetiva popularizar o desenvolvimento de ontologias, fortalecendo assim a pesquisa na área de ontologias na Ciência da Informação.

Nesse estágio, a pesquisa contém limitações, sendo a principal delas uma avaliação de usabilidade mais robusta que respalde o que a prática já tem mostrado. Nesse sentido, o editor tem sido testado por professores e alunos de graduação, mestrado e doutorado da Universidade Federal de Minas Gerais (UFMG) e da Universidade Federal de Juiz de Fora (UFJF), em disciplinas que envolvem ontologias e recuperação da informação. Além disso, o *Onto4AllEditor* tem sido testado, experimentalmente, no desenvolvimento de ontologias de base industrial em setores estratégicos, como a Companhia Energética de Minas Gerais (CEMIG). Espera-se, em breve, um retorno empírico no sentido de respaldar as aplicações.

Cabe ainda mencionar que alguém pode sugerir como limitação o fato de que os problemas apontados em outros editores, ao longo do texto, são de mais de cinco anos atrás. Contudo, as referências aqui apresentadas são as seminais da literatura e não se identificou nenhum estudo mais recente no processo de revisão. Ainda que o número de *plug-ins* de algumas ferramentas tenha aumentado nesse período, acredita-se que tal aumento não reflete necessariamente em melhorias significativas no uso dos editores de ontologia por profissionais e pesquisadores da Ciência da Informação.

Nem linguagens de modelagem, como a UML, nem linguagens lógicas, como o OWL e sua capacidade de raciocínio, figuram na formação e nas habilidades da maioria de estudantes e pesquisadores da CI. O *Onto4AllEditor* se justifica como forma de permitir acesso ao tema e mesmo disseminar a construção de ontologias na área. Também se justifica como uma ferramenta leve para o uso entre não-iniciados de outras áreas, como da Linguística, e para especialistas de domínio.

Conclui-se enfatizando que o principal propósito do *Onto4AllEditor* é ser uma ferramenta simples e acessível para construção de ontologias, de forma a

popularizar a atividade entre desenvolvedores não familiarizados com questões técnicas, lógicas e filosóficas. Como trabalhos futuros, está prevista a inclusão de funcionalidades para: a) extração de termos a partir de documentos utilizando processamento de linguagem natural (PLN); b) suporte a *reasoners*; c) validador de expressões lógicas na inserção de axiomas; e d) funções de *merging* e alinhamento de ontologias desenvolvidas no editor. Os recursos futuros serão incorporados na ferramenta em forma de *plug-ins*. Está previsto também um pré-cadastro de classes e relações ontológicas definidas em ontologias de alto e médio nível, em particular a BFO, IAO e IOF, de forma que os usuários possam se beneficiar da estrutura já criada nesses recursos.

Por fim, é importante destacar as instituições envolvidas na pesquisa e as tecnologias usadas na implementação. O *Onto4AllEditor* foi desenvolvido no Laboratório de Aplicações e Inovação em Computação⁴ da Universidade Federal de Juiz de Fora em parceria com o Programa de Pós-Graduação em Gestão e Organização do Conhecimento da Universidade Federal de Minas Gerais. Quanto às tecnologias usadas, o *Onto4AllEditor* é desenvolvido no *framework Laravel* versão 6.x, utilizando arquitetura *Model View Controller* (MVC) e as linguagens de programação PHP (*back-end*), HTML, CSS e Javascript (*front-end*) com o pacote *Laravel-AdminLTE*, mais um banco de dados MySQL. Em *JavaScript*, adotaram-se duas bibliotecas específicas: a) o *mxGraph*⁵, modificada para permitir a criação de um componente de diagramação interativo; b) a *API JQuery*, utilizada nas funcionalidades dinâmicas do editor, tais como a geração de alertas. O gerenciamento e controle de versão do código-fonte é feito através do *GitHub*.

Referências

ALATRISH, E. Comparison Some of Ontology Editors. **Management Information Systems**, [s. 1], Vol. 8, No. 2, pp. 018-024, 2013.

ALMEIDA, M.B. Revisiting ontologies: A necessary clarification. **Journal of the American Society for Information Science and Technology**, [s. 1], v. 64, n. 8, p. 1682-93, 2013.

ALMEIDA, M.B. **Ontologias em Ciência da Informação**, v. 1 – Teoria e Prática. Curitiba: CRV, 2020.

BITTNER, T.; DONNELLY, M. Logical properties of foundational relations in bio-ontologies. **Artificial Intelligence in Medicine**, [s. 1], v. 39, n. 3, p. 197–216, 2007.

BLACKBURN, S. **The Oxford dictionary of philosophy**. Oxford; New York: Oxford University Press, 1996.

BRAUN G., ESTEVEZ E. and FILLOTRANI, P. A Reference Architecture for Ontology Engineering Web Environments. **Journal of Computer Science & Technology**, [s. 1], v. 17, n. 2, p. 1–3, 2018.

CAÑAS, A. J., HILL, G., CARFF, R., SURI, LOTT, J., GÓMEZ, G., ESKRIDGE, T. C., ARROYO, M., CARVAJAL, R. CMapTools: a Knowledge Modeling and Sharing Environment. **First Int. Conference on Concept Mapping**, Madrid, Espanha, 2004.

CEUSTERS, W., SMITH, B., KUMAR, A.; DHAEN, C. Mistakes in medical ontologies: where do they come from and how can they be detected? **Stud. Health Technol. Inform.**, [s. 1], v. 1, n. 102, p. 145–164, 2004.

DAY-RICHTER J., HARRIS M. A., HAENDEL M. The Gene Ontology OBO-Edit Working Group, Suzanna Lewis, OBO-Edit—an ontology editor for biologists. **Bioinformatics**, [s. 1], v. 23, n. 16, p. 2198–2200, 2007.

DEGEN, W.; HELLER, B.; HERRE, H.; SMITH, B. GOL: Toward an axiomatized upper level ontology. In: **Proceedings of the 2nd Int. Conf. on Formal Ontology in Information Systems**. New York, USA: ACM, 2001. P.34–46.

GANGEMI, A., CATENACCI, C., CIARAMITA, M., LEHMANN, J. Modelling Ontology Evaluation and Validation. **Sure Y., Domingue J. (eds) The Semantic Web: Research and Applications**. ESWC 2006. Lecture Notes in Computer Science, v. 4011. Springer, Berlin, Heidelberg, 2006.

GÓMEZ-PÉREZ, A.; FERNÁNDEZ, M.; VICENTE, A. J. Towards a method to conceptualize domain ontologies. **ECAI WORKSHOP ON ONTOLOGICAL ENGINEERING**, Budapest, Hungria, 1996.

GRENON, P.; SMITH, B.; GOLDBERG, L. **Biodynamic Ontology: Applying BFO in the Biomedical Domain**. In: PISANELLI (ed.), **Ontologies in Medicine**, Amsterdam: IOS Press, p. 20–38, 2004.

GUARINO, N. **Formal Ontology in Information Systems (FOIS)**, Trento, Itália, 1998. Disponível em: <http://citeseer.ist.psu.edu/guarino98formal.html>. Acesso em: 2 jul. 2020.

GUARINO N., WELTY, C.A. An Overview of OntoClean. **Handbook on Ontologies**. Int. Handbooks on Information Systems. Berlin: Springer, 2004.

GUIZZARDI, G. **Ontological Foundations for Structural Conceptual Models**, 2005. Disponível em:

https://ris.utwente.nl/ws/portalfiles/portal/6042428/thesis_Guizzardi.pdf.

Acesso: 06 abr. 2020.

INDUSTRIAL ONTOLOGIES FOUNDRY (IOF). **Helpful materials on ontologies**, 2020. Disponível em: <https://www.industrialontologies.org/>. Acesso em: 20 mar. 2020.

KALYANPUR, A., PARSIA B., SIRIN E., GRAU B. C.; HENDLER, J. Swoop: A web ontology editing browser. **Web Semantics: Science, Services and Agents on the World Wide Web**, [s. 1], n. 4.2, pp. 144-153, 2006.

KETT, C.; ARTALE, A. Representing and reasoning over a taxonomy of part-whole relations. **Applied Ontology** 3, [s. 1], v. 1, n. 2, p. 91-110, 2008.

MALIK, Z. H. Usability Evaluation of Ontology Engineering Tools. **Computing Conference 2017**, London, UK.

MENDONÇA, F. M. **OntoForInfoScience: metodologia para construção de ontologias pelos cientistas da informação – uma aplicação prática no desenvolvimento da ontologia sobre componentes do sangue humano (Hemonto)**. Tese de Doutorado. Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brasil, 2015.

MENDONÇA, F. M.; ALMEIDA, M. B. Hemocomponents and Hemoderivatives Ontology (HEMONTTO): an Ontology About. **6º Seminário de Pesquisas em Ontologias do Brasil (ONTOBRAS)**, Belo Horizonte, MG, v. 1. p. 11-23, 2013.

MENDONÇA, F. M.; SOARES, A. L. Construindo ontologias com a metodologia *OntoForInfoScience*: uma abordagem detalhada das atividades do desenvolvimento ontológico. **CIÊNCIA DA INFORMAÇÃO (ONLINE)**, Brasília, DF, v. 46, p. 43-59, 2017.

MENDONÇA, F. M.; ZAIDAN, F. H. Ontologias para organização da informação em processos de transformação digital. **EM QUESTÃO**, Porto Alegre, RS, v. 25, p. 295-320, 2019.

MICHAEL, E.; WATERFELD, W. Overview of the Neon Toolkit. **Ontology Engineering in a Networked World**, [s. 1], pp. 281-301, 2012.

MUNN, K; SMITH B. **Applied Ontology: An Introduction**. Heusenstamm, Germany: Ontos Verlag, 2008.

MUSEN, M.A. The Protégé project: A look back and a look forward. AI Matters. **Association of Computing Machinery SIG-AI**, [s. 1], v. 1, n. 4, 2015.

NOY, N. F.; McGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology**. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, California, USA, 2001.

POVEDA-VILLÁLON, M.; SUÁREZ-FIGUEROA, M.C.; GÓMEZ-PÉREZ, A. A Double Classification of Common Pitfalls in Ontologies. **Proceedings OntoQual 2010 - Workshop on Ontology Quality at the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW)**, Lisboa, Portugal. p. 1- 12, 2010.

POVEDA-VILLÁLON, M.; SUÁREZ-FIGUEROA, M.C.; GÓMEZ-PÉREZ, A. Validating ontologies with oops! **International conference on knowledge engineering and knowledge management**, Springer, Berlin, Heidelberg., pp. 267-281, 2012.

RECTOR, A., DRUMMOND, N., HORRIDGE, M., ROGERS, J., KNUBLAUCH, H., STEVENS, R., WANG, H., WROE, C. Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns. **Proceedings of EKAW 2004**, Springer, Berlin, Heidelberg, pp: 63–81, 2004.

ROUSSEY, C., CORCHO, O., VILCHES-BLÁZQUEZ, L. M. A catalogue of OWL ontology antipatterns. **Proceedings of the fifth international conference on knowledge capture**, [s. 1.], pp. 205-206, ACM, 2009.

SALES, T. P.; GUIZZARDI, G. Ontological anti-patterns: Empirically uncovered error-prone structures in ontology-driven conceptual models. **Data & Knowledge Engineering**, [s. 1.], n. 99, p. 72-10, 2015.

SCHALLEY, A.C. Ontologies and ontological methods in linguistics. **Lang Linguist Compass**, [s. 1.], v. 1, 2019.

SCHULZ, S.; KUMAR, A.; BITTNER, T. Biomedical ontologies: what part-of is and isn't. **Journal of Biomedical Informatics**, [s. 1.] v. 39, p. 350-361, 2006.

SIRICHAOEN, W. V. Ontology Editors Approach for Ontology Engineering. **International Conference on Control, Automation and Robotics**, [s. 1.], 2018.

SMITH, B. **Ontology and Informations Systems**, 2004. Disponível em: <http://www.ontology.buffalo.edu/ontology>. Acesso em: 25 jan. 2016.

SOERGEL, D. **Functions of a Thesaurus / Classification / Ontological Knowledge Base**. College of Library and Information Services, University of Maryland, 1997.

SUÁREZ-FIGUEROA, M. C. (2008). **NeOn D 5.4.1. NeOn Methodology for Building Contextualized Ontology Networks**. NeOn project. Disponível em: <http://www.neon-project.org>. Acesso em: 06 abr. 2020.

VICKERY, B. C. Ontologies. **Journal of Information Science**, London, UK, v. 23, n. 4, p. 277-286, 1997.

VIGO, M.; BAIL, S.; JAY, C. STEVENS, R. Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. **International Journal of Human-Computer Studies**, [s. 1.], v. 72, n. 12, p. 835-845, 2014.

W3C 2020. **Ontology editors**. Disponível em: https://www.w3.org/wiki/Ontology_editors. Acesso em: 06 de Abril de 2020.

WARREN, P. **Ontology Users' Survey – Summary of Results**. The Knowledge Media Institute (KMi), [s. 1.], 2013.

Onto4AllEditor: a web-based graphic editor for building ontologies addressed to all kinds of information users

Abstract: Ontologies are knowledge representation artifacts used in the scope of Information Science, mainly in the field of Knowledge Organization, as well as in related areas. Even though the theme has been disseminated for more than two decades, the construction of ontologies still poses challenges for professionals, researchers and students of Information Science. Such challenges are mainly related to the logical-philosophical jargon required for the knowledge formalization stage and to the complexity of the available editors focused on computing. This research aims to provide an alternative to such challenges. We present here the *Onto4AllEditor*, a graphical, collaborative and web-based ontology editor, which includes simple and intuitive features. Based on the *OntoForInfoScience* methodology, the editor was designed to facilitate the work of users with distinguished levels of experience, with the aim of popularizing the construction of ontologies. We discuss the limitations of the popular editors available and compares them to which is offered by *Onto4AllEditor*. Finally, we conclude that *Onto4AllEditor*, even already including some advanced features, simplify the construction of ontologies, in particular, for those users who are starting in this type of activity.

Keywords: Ontology editor. Ontology construction. Knowledge representation.

Recebido: 22/07/2020

Aceito: 10/10/2020

Declaração de autoria

Concepção e elaboração do estudo: Fabrício Martins Mendonça, Lucas Piazzzi de Castro, Jairo Francisco de Souza, Maurício Barcellos Almeida, Eduardo Ribeiro Felipe

Coleta de dados: Fabrício Martins Mendonça, Lucas Piazzzi de Castro, Jairo Francisco de Souza, Maurício Barcellos Almeida, Eduardo Ribeiro Felipe

Análise e discussão de dados: Fabrício Martins Mendonça, Lucas Piazzzi de Castro, Jairo Francisco de Souza, Maurício Barcellos Almeida, Eduardo Ribeiro Felipe

Redação e revisão do manuscrito: Fabrício Martins Mendonça, Lucas Piazzzi de Castro, Jairo Francisco de Souza, Maurício Barcellos Almeida, Eduardo Ribeiro Felipe

Como citar

MENDONÇA, Fabricio Martins; CASTRO, Lucas Piazzzi; SOUZA, Jairo Francisco; ALMEIDA, Maurício Barcellos; FELIPE, Eduardo Ribeiro. Onto4AllEditor: um Editor Web Gráfico para Construção de Ontologias por Todos Tipos de Usuários da Informação. **Em Questão**, Porto Alegre, v. 27, n. 3, p.401 – 430, 2021. DOI: <http://doi.org/10.19132/1808-5245273.401-430>



1 Disponível em: <https://www.onto4alleditor.com/>. Acesso em: 29 jun. 2020.

2 Disponível em: https://www.w3.org/wiki/Ontology_editors. Acesso em: 30 jun. 2020.

3 Disponível em: <https://www.industrialontologies.org/>. Acesso em: 30 jun. 2020.

4 Disponível em: <http://www.ufjf.br/lapic/>. Acesso em: 18 mai. 2020.

5 Disponível em: <https://github.com/jgraph/mxgraph>. Acesso em: 18 mai. 2020.