Onto4AllEditor: um Editor Web Gráfico de Ontologias Direcionado a Diferentes Tipos de Desenvolvedores de Ontologias

Fabrício M. Mendonça[1], Lucas P. de Castro[1], Jairo F. de Souza[1], Maurício B. Almeida[2], Eduardo R. Felipe[2]

[1]Departamento de Ciência da Computação - Universidade Federal de Juiz de Fora (UFJF) - Caixa Postal 20010 – CEP 36016-970 – Juiz de Fora - MG - Brasil

[2]Escola de Ciência da Informação - Universidade Federal de Minas Gerais (UFMG) - Belo Horizonte - MG - Brasil

{fabricio.mendonca,lpiazzi,jairo.souza}@ice.ufjf.br,{mba,erfelipe}@ufmg.br

Abstract. *Literature surveys reveal several methodologies for building ontologies. A common step among all of them is the use of editors for organizing and formalizing knowledge. For researchers and students in Information Science, as well as for domain experts, the use of such editors is still an obstacle, because their majors does not favor the learning of relevant notions for the ontology development. Within this context, we have developed Onto4AllEditor, a graphical ontology editor, incorporated into a methodology, Web-oriented, which aims to provide functionalities for creating lightweight ontologies. Onto4AllEditor is part of an ongoing project whose main goal is to foster the development of ontologies and then reinforce the field's research.*

Resumo. *Levantamentos na literatura revelam diversas metodologias para construção de ontologias.Uma etapa comum a todas elas é o uso de editores para organizar e formalizar conhecimento. Para pesquisadores e estudantes da Ciência da Informação, bem como para especialistas de domínio, o uso de editores ainda é um empecilho pois sua formação não favorece o aprendizado de noções que são relevantes para desenvolver ontologias. Nesse contexto, desenvolveu-se o Onto4AllEditor, um editor gráfico de ontologias, orientado a Web, com metodologia incorporada, que objetiva fornecer funcionalidades para a criação de ontologias lightweight. O Onto4AllEditor é parte de um projeto em andamento que visa popularizar o desenvolvimento de ontologias e fortalecer a pesquisa na área.*

1. Introdução

A partir dos anos 1990, verificou-se um grande número de iniciativas para o uso de ontologias em sistemas de informação. Apontava-se a inconsistência das práticas nos primeiros anos de modelagem como a principal causa dos problemas de falta de integração entre sistemas [Smith e

Welty 2001]. Ainda que massivamente adotadas nos últimos 15 anos, ontologias já eram citadas nos anos 1960 [Mealy 1967]. Contam-se também iniciativas significativas nos anos 1980 [Wand, Storey e Weber 1999] e nos anos 1990 [Gruber 1992] [Guarino 1998], até a popularização nos anos 2000.

De fato, a modelagem conceitual é a atividade de descrever formalmente aspectos do mundo físico e social de interesse ao nosso redor para fins de compreensão e comunicação [Mylopoulos 1992]. Ao longo de todos esses anos, a criação de modelos conceituais foi motivada pela busca de melhorias na representação em sistemas de informação. A ontologia se presta a tais melhorias, pois é "[…] uma especificação conceitual que descreve conhecimento sobre um domínio de uma forma independente dos estados epistêmicos e dos estados das coisas" [Guizzardi 2005, p. 83].

Por todo esse histórico, o uso de ontologias tem lugar principalmente nas áreas da Modelagem Conceitual e de Representação do Conhecimento. Entretanto, outras áreas científicas também fazem uso de ontologias, por exemplo, a Ciência da Informação [Almeida 2013] e a Linguística [Schalley 2019]. A Ciência da Informação (CI) herdou métodos de classificação e indexação, atividades centrias da área que, atualmente, estão relacionadas ao uso de ontologias. Por ser Ciência Social Aplicada, a CI se aproximou de teorias da Filosofia, o que resultou em deficiências práticas como o processo de construção de ontologias [Smith 2004]. Além disso, a formação da CI não privilegia requisitos relevantes para construir ontologias, como lógica, modelagem, dentre outros.

O uso de editores de ontologia populares como o *Protégé* [Munsen 2015] ainda é um desafio em CI. Apesar de ser uma ferramenta com mais de duas décadas de utilização e a melhor avaliada [Malik 2017] [Warren 2013] certos usuários ainda encontram dificuldades no uso de recursos que tal editor oferece. Não apenas no âmbito da CI, mas também em outras áreas que fazem uso de ontologias, alguns problemas comuns apontados em entrevistas com usuários do Protégé, Web Protégé, TopBraid e SWOOP são: i) necessidade de um ambiente mais colaborativo entre especialistas do domínio para modelagem da ontologia; ii) falta de suporte para visualização de ontologias; iii) feedback pobre na depuração de erros da ontologia; iv) falta de suporte para pesquisa de termos em ontologias externas para reuso; v) interface pouco amigável na navegação pela hierarquia de classes; dentre outros [Vigo 2014].

Nesse contexto, foi criado um projeto de pesquisa (em andamento) com o objetivo de disseminar o desenvolvimento de ontologias no âmbito da CI, ainda que não restrito apenas a essa área. A partir da metodologia de construção de ontologias de Mendonça (2015) – denominada *OntoForInfoScience* – desenvolveu-se o *Onto4AllEditor*[1], um editor de ontologias gráfico, disponível via *Web*, que fornece funcionalidades básicas para a criação de ontologias e inclui recursos de suporte à problemas apontados nos editores atuais. Exemplos desses recursos são: i) interface gráfica e colaborativa disponível na web; ii) associação entre etapas da metodologia de construção e tarefas executadas no editor; iii) console de avisos e erros de modelagem; iv) reuso de ontologias de alto nível; v) geração de relatórios das ontologias; dentre outros.

1 Disponível em: https://www.onto4alleditor.com/. Acesso em: 21 de maio de 2020.

O *Onto4AllEditor* tem como propósito principal popularizar a construção de ontologias, especialmente, entre usuários com pouca ou nenhuma experiência em ontologias. Permite a construção de ontologias de qualquer tipo no editor, mas o foco são as ontologias *lightweight*, mais comuns entre usuários menos experientes. O público-alvo da ferramenta envolve, principalmente, profissionais da CI, Ciência da Computação, Linguística e especialistas de domínio, embora possa ser utilizado por profissionais de qualquer área e mesmo por usuários com experiência em ontologias. Sobre as linguagens de representação, a versão atual do *Onto4AllEditor* trabalha com a *Ontology Web Language (OWL)* nas funções de importação e exportação.

O restante do presente artigo está organizado como segue: a Seção 2 traz um *background*, destacando problemas na construção de ontologias e descrevendo de forma breve a metodologia *OntoForInfoScience*; a Seção 3 lista editores de ontologia e apresenta trabalhos relacionados; a Seção 4 descreve o *Onto4AllEditor* em termos de suas características básicas, funcionalidades e usabilidade; e, finalmente, a Seção 5 oferece considerações finais e trabalhos futuros.

2. Background

Nesta seção apresenta-se *background* teórico destacando aspectos que motivaram a iniciativa do editor *Onto4AllEditor*. A seção 2.1 descreve problemas conhecidos na construção de ontologias, enquanto a Seção 2.2 traz uma breve descrição da metodologia que fundamenta o editor.

2.1 Dificuldades na Construção de Ontologias

O processo de construção de ontologias é complexo e oneroso, além de existirem diferentes tipos de ontologia, cada uma com um propósito específico. Por esse motivo, é importante destacar, brevemente, os tipos de ontologia e usos recomendados, antes de discorrer sobre dificuldades encontradas na construção de ontologias.

Existem diferentes classificações para tipos de ontologias na literatura da área. Ao presente artigo interessa destacar dois critérios: (i) estrutura e propósito; (ii) grau de formalismo. No primeiro critério, ontologias são classificadas como [Guarino 1998] [Haav e Lubi 2001] [Guizzardi 2005]: (a) *ontologias de alto nível, meta-ontologias ou ontologias de fundamentação*: que descrevem classes gerais, independentes do problema ou domínio, tais como: espaço, tempo, matéria, objeto, evento, ação, entre outros; (b) *ontologias de domínio*: que descrevem o vocabulário de um domínio específico, por exemplo, multimídia, patrimônio cultural, medicina ou automóveis; (c) *ontologias de tarefa*: que descrevem uma tarefa ou atividade, por exemplo, diagnósticos ou compras, através da inserção de termos especializados; (d) *ontologias de aplicação:* descrevem entidades dependentes de um domínio ou tarefa em particular, que correspondem à funções desempenhadas por entidades do domínio durante a execução de uma determinada atividade. Quanto ao critério grau de formalismo, as ontologias são classificadas como [Ding e Engels 2001] [Gómez-Pérez 2004]: (a) ontologias leves *(light-weight):* aquelas com pouco rigor formal, geralmente, composta por classes facilmente compreensíveis e de relações usuais entre elas, não incluindo, por exemplo, relações e axiomas lógicos; (b) ontologias

pesadas *(heavy-weight)*: se referem às ontologias com alto rigor formal, incluindo além das classes e relações comuns, relações especiais e alto grau de axiomatização, baseado nas linguagens lógicas, tal como a lógica de primeira ordem.

Apesar da quantidade de ontologias desenvolvidas ao longo dos últimos 20 anos, a literatura da área aponta erros comuns de desenvolvimento. O levantamento apresentado aqui não cobre todos os tipos de erros – pois seriam dezenas – mas apenas introduz tipos comuns. Trabalhos bem conhecidos revelam denominações e abordagens diversas: erros e padrões comuns [Rector et al 2004]; anomalias ou *pitfalls* [Poveda, Figueroa e Perez 2010, 2012]; anti-padrões ontológicos [Roussey, Corcho e Blázquez 2009] [Sales e Guizzardi 2015]; erros de definições de classes e relações [Ceusters el al 2004] [Munn e Smith 2008]; sobrecarga de relações [Guarino e Welty 2004], dentre outros. Além disso, encontram-se propostas de classificação dos erros [Gangemi et al 2006] [Poveda, Figueroa e Perez 2010], as quais, contudo, não parecem consensuais. Assim, enquanto são encontrados na literatura extensas listas de erros – por exemplo em [Gómez-Pérez e Suárez-Figueroa 2009] – a lista abaixo não é exaustiva, mas cobre a maioria dos tipos de erros:

- Erros que envolvem relações:
  - Uso de relações não genuinamente ontológicas [Ceusters el al 2004];
  - Sobrecarga de relações é-um [Guarino e Welty 2004]
  - Inclusão de relações não neutras em ontologia de alto nível [Munn e Smith 2008];
  - Uso relações todo-parte inconsistentes [Bittner e Donnely 2007] [Keet e Artale 2008];
  - Uso incorreto de inversos [Poveda-Villalon, Suárez-Figueroa e Gómez-Perez 2010];
- Erros que envolvem classes:
  - Definições imprecisas, recursivas. [Munn e Smith 2008], [Sales e Guizzardi 2015];
  - Confusão instância-classe [Noy e McGuinnes 2001], [Schulz, Bittner e Kumar 2006];
  - Classes com sinônimos [Poveda-Villalon, Suárez-Figueroa e Gómez-Perez 2010];
  - Uso de “classes miscelâneas” [Sales e Guizzardi 2015];
- Erros conceituais e lógicos:
  - Criação de polissemias [Poveda-Villalon, Suárez-Figueroa e Gómez-Perez 2010];
  - Confusão uso-menção [Munn e Smith 2008];
  - Uso de herança múltipla [Munn e Smith 2008], [Noy e McGuinnes 2001];
  - Uso equivocado do “*and*” e “*or*” lógicos [Roussey, Corcho e Blázquez 2009];
  - Erros “*some not*” e “*not some*” [Rector et al 2004] [Roussey, Corcho e Blázquez 2009];
  - Uso do “*Relator Mediating Overlapping*” [Sales e Guizzardi 2015];

- Uso do "*SumOfSome*" [Roussey, Corcho e Blázquez 2009];
- Uso de "*UniversalExistence*"[Rector et al 2004] [Sales e Guizzardi 2015].

Cabe ainda citar padrões e boas práticas para minimizar erros, cujos exemplos são: *Ontology Design Patterns* (ODPs), que incorporam *templates* de sucesso [Gangemi e Presutti 2009]; e regras de modelagem indutivas, que envolvem a derivação de regras a partir de ODP's [Guizzardi, Graças e Guizzardi 2011]. Exemplos de recomendações e boas práticas são:

- uso de definições claras e concisas; uso de classes primitivas e definidas [Rector et al 2004];
- inclusão de relações genuinamente ontológicas; evitar o uso de herança múltipla [Munn e Smith 2008];
- inclusão de relações inversas; definição de todas propriedades textuais de classes e relações; não utilização de definições recursivas [Michael e Waterfeld 2012];
- regras indutivas derivadas três tipos de ODP´s: padrão de fase, padrões de subtipo, padrão de modelagem de papéis [Guizzardi, Graças e Guizzardi 2011].

Apesar de todo esse esforço, as dificuldades mantêm-se no processo de construção de ontologias. Para algumas delas, a presente pesquisa busca fornecer recursos envolvendo a metodologia e o editor propostos, conforme é explicado nas Seções 4 e 5.

2.2 Breve Visão da Metodologia de Construção de Ontologias

Existem metodologias de construção de ontologia desde os anos de 1990. Exemplos são: *Methontology* [Gómez-Perez, Fernandez-Lopes e Vicente 1996]; *Método 101* [Noy e McGuiness 2001]; *NeOn* [Suaréz-Figueroa 2008]; *Up for ONtology* (UPON) [De Nicola, Missikoff e Navigli 2009]), *Systematic Approach to Build Ontologies* (SABiO) [Falbo 2014], para citar algumas. As metodologias, apesar de bem estabelecidas, convivem com erros (vide Seção 2.1) e baixa qualidade de ontologias disponíveis na Web [Vigo 2014].

O projeto, bem como a concepção do *Onto4AllEditor*, partiu de dificuldades conhecidas e de necessidades da área de CI. Por esse motivo, optou-se por adotar a metodologia para construção de ontologias *OntoForInfoScience* [Mendonça 2015]. Tal metodologia contém, como diferencial em relação a outras, a tentativa de acessível o ciclo de desenvolvimento ontológico ao explicar termos técnicos, questões lógicas e filosóficas envolvidas, as quais raramente são de conhecimento ou domínio dos iniciantes interessados. De fato, dificuldades no desenvolvimento ocorrem porque as metodologias contém passos bem estabelecidos, mas não detalhados [Mendonça e Almeida 2016]. A utilização da *OntoForInfoScience* no projeto do editor contribui para tornar mais acessível a construção de ontologias para usuários menos experientes.

A *OntoForInfoScience* incentiva o reuso de ontologias a partir de ontologias de alto nível e de domínio. Por isso mesmo, as etapas da *OntoForInfoScience* são baseadas no melhor de outras metodologias, reutilizando partes de três metodologias conhecidas na literatura:

Methontology, *NeOn e Método 101*. Prescreve um conjunto de oito etapas metodológicas (1-Especificação; 2-Aquisição de Conhecimento; 3-Conceitualização; 4-Fundamentação ontológica; 5-Formalização da ontologia; 6-Avaliação; 7-Documentação; 8-Disponibilização) e uma etapa que avalia a necessidade de ontologia ou tesauro. Com exceção da etapa 2, todas as demais podem ser realizadas no *Onto4AllEditor*. Dessa forma, busca-se agilizar a modelagem ontológica e orientar o profissional da informação nesse processo, minimizando erros e dificuldades.

3. Editores de Ontologia: Exemplos e Problemas

Como mencionado, acredita-se que parte dos problemas de construção de ontologia podem ser creditados a limitações de editores. A presente seção lista editores conhecidos: enquanto a Seção 3.1 traz editores disponíveis na *Web*, a Seção 3.2 apresenta comentários e avaliação sobre o uso dos editores.

3.1. Uma Visão Geral sobre Editores de Ontologia

Essa seção lista, de forma não exaustiva, editores citados em três referências atualizadas (Tabela 1), à saber: [W3C 2020] [IOF 2020] [Braun, Estevez e Fillottrani 2019].

Tabela 1. Lista de editores de ontologias (não exaustiva, ordem alfabética)

Editor	Características	Referência
CmapTools	Editor de mapeamento conceitual. Permite representar nodos e mapas conceituais em ambiente gráfico.	https://cmap.ihmc.us/cmaptools/
Eddy for Graphol	Editor que usa a linguagem gráfica Graphol. Desenvolve grafos compostos por nós e conexões evitando textos.	https://www.obdasystems.com/eddy
Graffo	Editor de código fonte aberto para a apresentação de classes, propriedades e restrições em OWL.	https://essepuntato.it/graffoo/
Hozo	Editor, servidor e gerenciador de ontologias em ambiente distribuído; baseado na Web para colaboração.	http://www.hozo.jp/
ICOM	Permite manipular múltiplas ontologias; interface gráfica com suporte à digrama de classes e raciocinador.	https://www.inf.unibz.it/~franconi/icom/
Menthor	Editor de código aberto, permite integração com a OntoUML, estereótipos e importação do *Ent. Architect*.	https://ontouml.org/ontouml/tooling/
NeOn Toolkit	Inclui metodologia para desenvolvimento de ontologias em larga escala em ambiente distribuído.	http://neon-project.org/
NORMA	Editor para Object-Role Modeling (ORM) e ontologias com suporte para raciocínio automático.	Software não disponível
OBO-Edit	Editor de código aberto, codificado em Java. Otimizado para o formato ontológico OBO.	http://oboedit.org/
OLED	Editor baseado na linguagem OntoUML, fundamentada na	https://ontouml.org/ontouml/

	UFO; oferece validação, simulação e detecção de padrões.	tooling/
OWLGrEd	Interface gráfica para editar e visualizar ontologias, extensão por plugins inclusive para modelagem UML.	http://owlgred.lumii.lv/
Protégé	Editor de código aberto com interface para criação e edição; extensão por plugins mais processador de consultas.	https://protege.stanford.edu/
SWOOP	Editor de código aberto com suporte a OWL de arquitetura baseada em Web. Possui suporte de raciocínio automático.	http://www.mindswap.org/
TopBraid Composer	IDE comercial para modelagem RDF em interface gráfica. Tem capacidade de inferência, mapeamento e consultas.	https://www.topquadrant.com/products/topbraid-composer/
yEd Graph Editor	Editor múltiplo comercial, permite trabalhar com grafos de redes semânticas e OWL.	https://www.yworks.com/products/yed
WebProtégé	Versão colaborativa na Web do Protégé. Suporta edições em OWL e OBO); importa e exporta diversos formatos.	https://protege.stanford.edu/

3.2. Sobre a Utilização dos Editores de Ontologia

Para explorar comentários sobre editores de ontologia, realizou-se um levantamento, pesquisa não sistemática, o qual retornou duas dezenas de trabalhos relacionados aos editores de ontologias nos últimos dez anos. Pesquisas sobre a utilização de editores encontrados trazem impressões sobre o uso e aplicação de editores, entretanto, tais trabalhos não incluem todos editores citados na Seção 3.1. Foi possível constatar que a literatura da área ainda carece de pesquisas sobre utilização e avaliação de editores de ontologia, abarcando um número maior de ferramentas.

Pesquisa quantitativa com 65 profissionais que trabalham com ontologias identificou, dentre outras questões, quais os editores mais usados [Warren 2013]. O *Protégé* foi identificado como o mais utilizado com 50% dos entrevistados usando a ferramenta; o *TopBraid Composer*, um editor comercial, apareceu em segundo lugar com 14%; o *NeOn Toolkit* [Michael e Waterfeld, 2012], *CmapTools* [Cañas et al, 2004] e SWOOP [Kalyanpur et al, 2006] apareceram, respectivamente, com 6%, 5% e 4%; além dos editores voltados para Biomedicina, o *OBO Edit* [Day-Richter, Harris e Haendel, 2007] e *Neurolex*, com a 3% e 2%, respectivamente.

Pesquisa sobre os aspectos da usabilidade de editores [Malik 2017] apresentou aos participantes uma sequência de tarefas em cinco editores. Dentre os resultados, o *Protégé* apareceu como o mais fácil de usar e que exigiu menos tempo para as tarefas; o *IHMC CmapTool* apresentou os piores resultados. O tempo para as tarefas com o Protégé foi cerca da metade daquele gasto com outros 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, posição ocupada pelo *Protégé* a partir do *feedback* positivo dos participantes. O *NeOn Toolkit* ficou na terceira posição, SWOOP na quarta e *IHMC Cmap Tools* foi considerado a pior ferramenta dentre as avaliadas.

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 são: *NeOn Toolkit, SWOOP e OntoStudio*. Um dos aspectos mencionados por aqueles que escolheram o *Protégé* foi o tutorial (*pizza*), outros motivos incluem: sua grande comunidade de usuários, o fato de ser gratuito e ter código fonte aberto.

A partir dos editores de ontologia citados e pesquisas de avaliação, tornou-se possível levantar algumas necessidades à área de modelagem ontológica a serem incluídas em um editor. Nesse sentido, além de seu propósito principal de uma ferramenta para popularizar a construção de ontologias *lightweight* entre usuários com pouca ou nenhuma experiência, o *Onto4AllEditor* é um software gratuito e vem desenvolvendo funcionalidades para melhorar a colaboração na modelagem entre especialistas - através de uma interface dinâmica e intuitiva disponível na web -, fornecer um melhor feedback dos erros de modelagem, facilitar a pesquisa e reuso de ontologias de alto nível, além de guiar os usuários através de passos da metodologia de construção. Detalhes das características e funcionalidades mencionadas são apresentadas na próxima seção (Seção 4).

4. Onto4AllEditor: Características e Funcionalidades

Esta seção descreve o *Onto4AllEditor* apresentando características, funcionalidades e etapas para criação de ontologias. Cabe destacar que o editor *Onto4AllEditor* é precedido, em suas principais características, por iniciativas pioneiras como o *OntoUML Lightweight Editor* (OLED), que auxilia a criação de ontologias bem fundamentadas usando a linguagem OntoUML [Guerson et al 2015]. O OLED possui funcionalidades de suporte, como verificação sintática e validação de relações de parentesco para validar constructos OntoUML. Outro exemplo similar é o CROWD, um editor gráfico que emprega raciocínio lógico para verificar especificações e sugerir restrições [Braun el al 2018].

O *Onto4AllEditor* ainda assim exibe diferenciais: além da interface gráfica, o editor é capaz de apontar erros de modelagem em tempo real. A estratégia foi implementar um console de avisos, seguindo a ideia dos console de erros das IDE´s de programação. Ao contrário dos erros detectados por compiladores, o *Onto4AllEditor* alerta para erros metodológicos, os quais, ainda assim não são de correção obrigatória. Os avisos implementados até o momento são apresentados adiante nesta seção.

Nem linguagens de modelagem, como UML, nem linguagens lógicas, como OWL e sua capacidade de raciocínio, figuram na formação e 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, especialmente, ontologias. Também se justifica como uma ferramenta leve entre não-iniciados de outras áreas, como da Linguística e especialistas de domínio. Sobre a linguagem de representação formal do editor é utilizado

OWL/XML e OWX. O *Onto4AllEditor* tem como característica central a modelagem gráfica em uma interface intuitiva e responsiva (Figura 1).

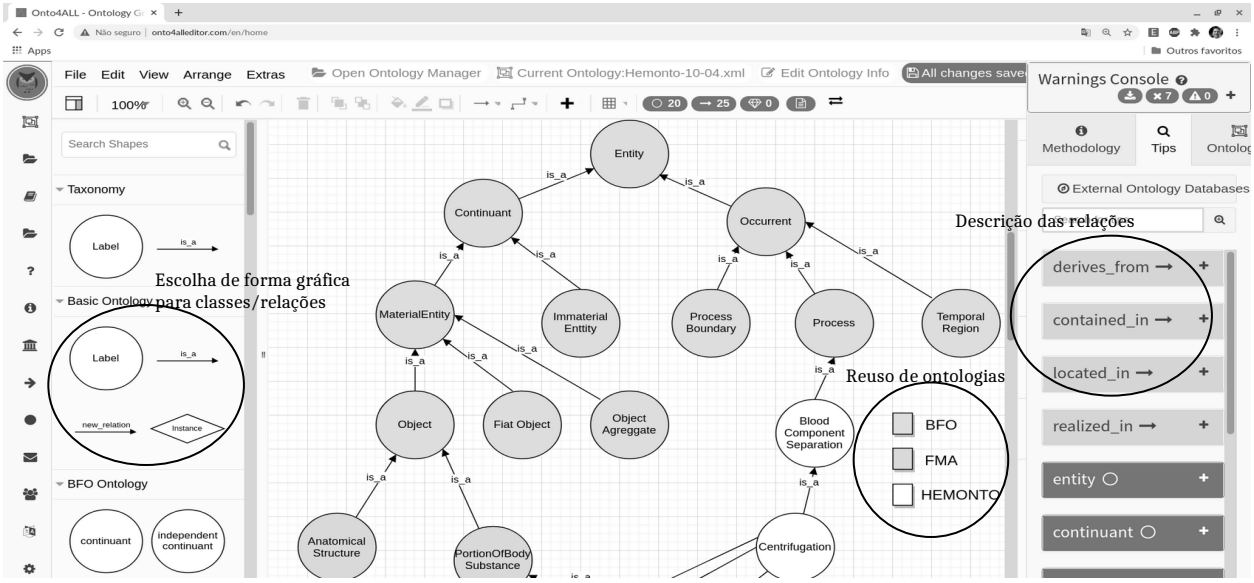

Figura 1. Interface de modelagem gráfica de ontologias no Onto4AllEditor

A Figura 1 mostra um fragmento da HEMONTO, uma ontologia sobre hemoterapia [Mendonça e Almeida 2016]. Na aba do menu superior ativa-se a interface gráfica, onde são inseridas classes, relações e propriedades. Cores e legendas podem ser usadas para indicar informações relevantes como, por exemplo, classes importadas de outras ontologias. A HEMONTO (Figura 1) importa classes da *Basic Formal Ontology (BFO)* e do *Foundational Model Anatomy (FMA).*

O reuso de ontologias é uma prática recomendada [Degen et al 2001] [Grenon e Smith 2004] que está presente tanto na *OntoForInfoScience* (etapa 4) quanto no *Onto4AllEditor*. No editor, o administrador pode cadastrar classes, relações e propriedades de ontologias de alto nível, para que os demais usuários possam reutilizar tais recursos em outras ontologias de domínio. Um exemplo são as relações *contained_in* e *derives_from*, provenientes da BFO e ilustradas na Figura 1. Dentre as funcionalidades, existe ainda a busca de classes e propriedades cadastradas no banco de dados do editor, que facilita o reuso.

O editor é capaz de realizar sete das oito etapas da *OntoForInfoScience*, as quais servem para orientar o desenvolvedor na construção da ontologia. O editor inclui caixas de diálogo (vide Figura 2) contendo descrição de cada etapa e as tarefas que devem ser executadas. A Figura 2 mostra a explicação da primeira etapa – especificação da ontologia – e como realizá-la a partir do *template* de especificação. A segunda etapa da metodologia é a única não realizada pelo *Onto4AllEditor*, pois envolve aquisição do conhecimento do domínio a partir de outras fontes de informação.

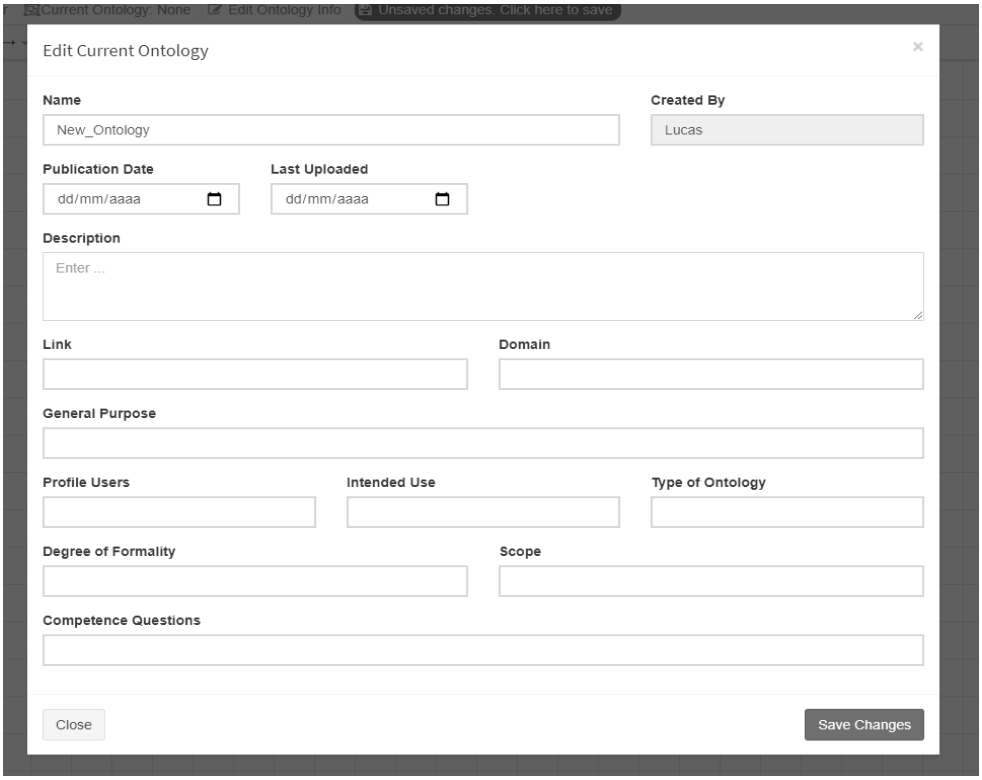

Figura 2. Etapa 1 da metodologia OntoForInfoScience no Onto4AllEditor.

As etapas 3, 4 e 5 são realizadas via comandos da interface, com funções que possibilitam a inserção de elementos obrigatórios, por exemplo: i) a inserção de classes, relações e instâncias; ii) a definição das propriedades das classes e relações; iii) axiomas com validação da sintaxe; iv) questões de competência (da etapa especificação). Cabe destacar que na etapa 5 – formalização da ontologia – o editor permite a inserção de construtores lógicos, tais como: uniões, interseções, negações, quantificadores existenciais e universais, através da funcionalidade "*Edit Properties*" de cada classe ou relação, a qual inclui atributos como *DisjointWith, EquivalentTo, SubClassOf,* entre outros.

A etapa de validação – etapa 6 – foi implementada em um console de avisos (Figura 3), que exibe alertas ao usuário sobre ações de modelagem que devem ser evitadas, além de contabilizar o número de classes, relações e instâncias já inseridos. Os avisos até então implementados são: i) classes duplicadas; ii) uso incorreto da relação "*instance_of*" entre classes; iii) herança múltipla; iv) circularidade; v) uso indevido de relações inversas; vi) falta de anotações e metadados; vii) erros no uso do ambiente gráfico.

A Figura 3 ilustra avisos no console para uma ontologia de pizza, onde o usuário recebeu 2 *warnings* (em destaque): i) herança múltipla na classe Mushroom-Pizza, que é subclasse, simultaneamente, de Pizza e *Veggie-Pizza*; ii) uso incorreto da relação "*instance_of*" entre as classes *Four-Cheese-Pizza* e *Pizza*, já que tal relação deve conectar uma instância à uma classe. Em ambos os casos, o console detecta dinamicamente esses problemas, informando o horário, a respectiva classe ou relação, além de um identificador do aviso. O relatório de avisos pode ser exportado a partir do console em um arquivo texto.

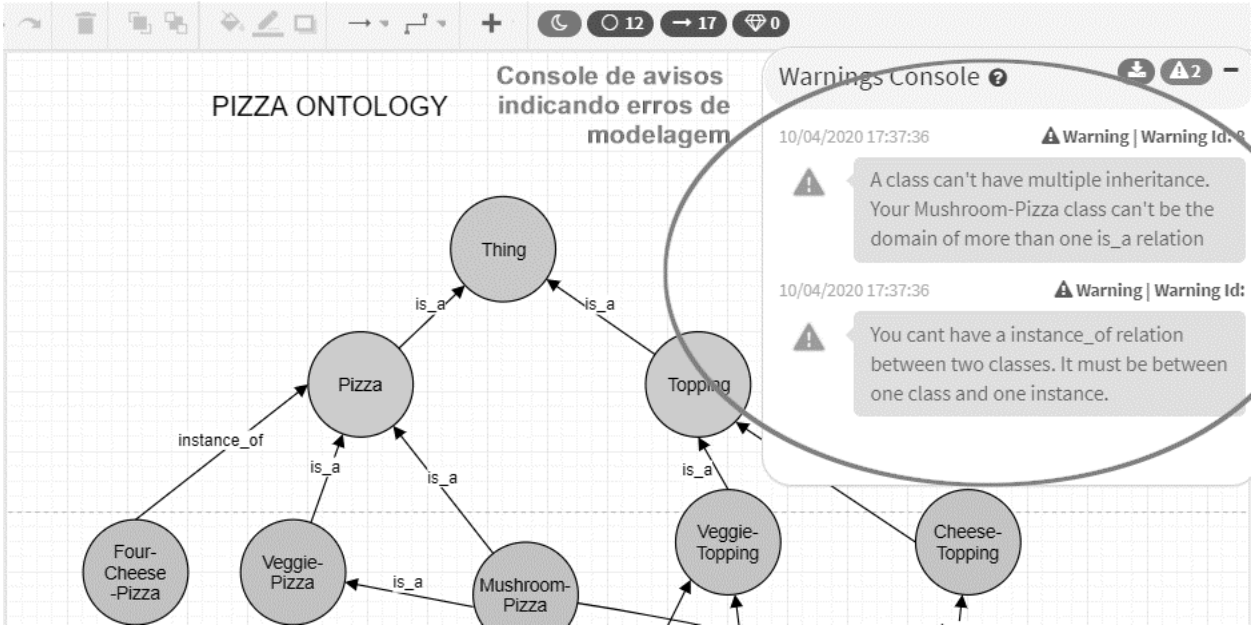

Figura 3. Console de avisos do Onto4AllEditor

A etapa de documentação – etapa 7 – é realizada no *Onto4AllEditor* ao longo de todo o processo de desenvolvimento. Além disso, o usuário pode acessar um repositório específico para gerenciar suas próprias ontologias. Essa funcionalidade exibe as dez últimas ontologias editadas, bem como as cinco favoritas. Por fim, a etapa de disponibilização – etapa 8 – procede a exportação para um dos três formatos disponíveis no editor: imagem (.svg), texto semi-estruturado (.xml) e lógica (.owl/xml ou .owx). A exportação para lógica permite que a ontologia possa ser manuseada em outros editores caso o usuário tenha essa necessidade. Em sua versão atual, o *Onto4AllEditor* utiliza os formatos OWL/XML e OWX para representação formal das ontologias, disponível nas funções de importação e exportação da ferramenta.

Por fim, é importante destacar as instituições envolvidas nesta pesquisa e as tecnologias usadas na implementação do editor. O *Onto4AllEditor* foi desenvolvido no Laboratório de Aplicações e Inovação em Computação[2] 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*, e banco de dados MySQL. Em *JavaScript*, adotaram-se duas bibliotecas específicas: i) o *mxGraph*[3], modificada para permitir a criação de um componente de diagramação interativo; ii) 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*.

2 Disponível em: http://www.ufjf.br/lapic/. Acesso em: 18 de maio de 2020.
3 Disponível em: https://github.com/jgraph/mxgraph. Acesso em: 18 de maio de 2020.

5. Considerações Finais

A presente pesquisa apresentou o editor web gráfico de ontologias – *Onto4AllEditor* – como alternativa para modelagem ontológica para não especialistas. O editor se fundamenta na *OntoForInfoScience*, uma metodologia de atividades do ciclo de desenvolvimento ontológico e direcionada para desenvolvedores menos experientes. Metodologia e editor fazem parte de um projeto em andamento que objetiva popularizar o desenvolvimento de ontologias e assim fortalecer a pesquisa na área, incluindo a Ciência da Informação.

Para cumprir seu propósito, o *Onto4AllEditor* faz uso de interface gráfica, dentre outros recursos que garantem a geração de conteúdo formal em OWL. Além da interface intuitiva, o editor destaca-se pelo console de avisos para erros. Na comparação com outros editores, destaca-se o uso de recursos gráficos e o acesso via Web, uma vez que a maioria dos editores com esses recursos têm alto custo de licença. Não procede comparar o *Onto4AllEditor* ao *Protégé*, visto que são ferramentas em estágios completamente diferentes, criadas em contextos e com recursos bastante díspares. Entretanto, por todos os motivos apresentados, com destaque para o uso por profissionais da CI, Linguística e especialistas de domínio, o *Onto4AllEditor* se justifica.

Esse estágio da pesquisa contém limitações, e uma das mais importantes é uma avaliação de usabilidade que respalde a que a prática já tem mostrado. Nesse sentido, o editor está sendo testado por professores e alunos de graduação, mestrado e doutorado de duas Universidades Federais em disciplinas que envolvem a construção de ontologias. Espera-se um retorno concreto em breve nesse sentido. Além disso, o *Onto4AllEditor* também tem sido testado, experimentalmente, no desenvolvimento de ontologias de base industrial em setores estratégicos.

Por fim, 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 inexperientes e não familiarizados com questões técnicas, lógicas e filosóficas. Como trabalhos futuros estão previstos a inclusão de funcionalidades para extração de termos a partir de documentos utilizando processamento de linguagem natural (PLN), o alinhamento de ontologias desenvolvidas no editor e um trabalho mais acurado de verificação de características desejáveis do editor. Enfatiza-se que o visualização é apenas um dos recursos relevantes, dentre muitos que precisam ser considerados, e os interessados podem consultar [Katifori, Halatsis, Lepouras et al. 2007]. 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.

6. Referências

Almeida M.B. (2013). Revisiting ontologies: A necessary clarification. Journal of the American Society for Information Science and Technology. 2013; 64(8):1682-93.

Bittner, T.; Donnelly, M. (2007). "Logical properties of foundational relations in bio-ontologies". In: Artificial Intelligence in Medicine, v. 39, n. 3, p. 197–216.

Braun, G., Gimenez C., Cecchi, L. and Fillottrani, P. (2016) “Towards a Visualization Process for Ontology-Based Conceptual Modelling.” In: *ONTOBRAS* 2016.

Braun G., Fillottrani, P., Cecchi, L., Gimenez, C., Oyarzun, A. (2018). Crowd v1.0: A Visual Web Tool for Ontology Engineering.

Braun G., Estevez E. and Fillottrani, P. (2018). “A Reference Architecture for Ontology Engineering Web Environments”. In: *Journal of Computer Science & Technology*, vol. 17, no. 2, pp. 1–3, 2018.

Cañas, A. J., Hill, G., Carff, R., Suri, Lott, J., Gómez, G., Eskridge, T. C., Arroyo, M., Carvajal, R. (2004) “CMapTools: a Knowledge Modeling and Sharing Environment”. First Int. Conference on Concept Mapping, Spain.

Ceusters, W., Smith, B., Kumar, A. e Dhaen, C. (2004) “Mistakes in medical ontologies: where do they come from and how can they be detected?” In *Stud. Health Technol. Inform. 102*, p. 145–164.

De Nicola, A.; Missikoff, M.; e Navigli, R. (2009) “A software engineering approach to ontology building”. In: *Information Systems* 34, p. 258–275, 2009.

Degen, W.; Heller, B.; Herre, H.; Smith, B. (2001). “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.

Ding, Y.; Engles, R. (2001). “IR and AI: Using Co-occurrence Theory to Generate Lightweight Ontologies”. In: *DEXA Workshop*, p. 961-965, 2001.

Falbo, R.D.A. (2014). SABiO: Systematic Approach for Building Ontologies. In ONTO. COM/ODISE@ FOIS.

Gangemi, A., & Presutti, V. (2009). “Ontology design patterns”. In *Handbook on ontologies* (pp. 221-243). Springer, Berlin, Heidelberg.

Gómez-Pérez, A.; Fernández, M.; Vicente, A. J. (1996) “Towards a method to conceptualize domain ontologies”. In: *ECAI WORKSHOP ON ONTOLOGICAL ENGINEERING*, 1996, Budapest, HUN.

Gómez-Pérez, A. (2004). “Ontology Evaluation”. In: Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems. Pp: 251-274. 2004.

Gómez-Pérez, A.; Suárez-Figueroa, M.C. (2009). NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology. Disponível em: <http://oa.upm.es/5475/1/INVE_MEM_2009_64399.pdf>. Acesso em: 07 de abril 2020.

Grenon, P.; Smith, B. (2004). “SNAP and SPAN: Towards Dynamic Spatial. Spatial Cognition & Computation”, v.4, n.1, p. 69-104. Disponível em: <http://ontology.buffalo.edu/smith/articles/SNAP_SPAN.pdf>. Acesso: 06 abril de 2020.

Gruber, T. (1992). What is an Ontology? Disponível em: http://www.ksl.stanford.edu/kst/what-is-an-ontology.html Acesso: 20 maio de 2020.

Guarino, N. (1998). Formal Ontology in Information Systems. Disponível em: <http://citeseer.ist.psu.edu/guarino98formal.html>. Acesso em: 06 de abril 2020.

Guarino N., Welty C.A. (2004). An Overview of OntoClean. In: Staab S., Studer R. (eds) Handbook on Ontologies. Int. Handbooks on Information Systems. Berlin: Springer.

Guerson, J.; Sales, T. P.; Guizzardi G., et al. (2015). “OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies,” In: *IEEE 19th International Enterprise Distributed Object Computing Workshop*, pp. 144-147, 2015.

Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. Disponível em: https://ris.utwente.nl/ws/portalfiles/portal/6042428/thesis_Guizzardi.pdf. Acesso em: 06 de abril de 2020.

Guizzardi, G., das Graças, A. P., & Guizzardi, R. S. (2011). “Design patterns and inductive modeling rules to support the construction of ontologically well-founded conceptual models in OntoUML”. In *International Conference on Advanced Information Systems Engineering* (pp. 402-413). Springer, Berlin, Heidelberg.

Haav, H.M.; Lubi, T.L. (2001). “A survey of concept-based information retrieval tools on the web”. Disponível em: <http://www.science.mii.lt/ADBIS/local2/haav.pdf>. Acesso em: 17 de Abril de 2020.

Industrial Ontologies Foundry (2020). Helpful materials on ontologies. Disponível em: https://www.industrialontologies.org/. Acesso em: 20 de abril 2020.

Kalyanpur A., Parsia B., Sirin E., Grau B. C. and Hendler, J. (2006). "Swoop: A web ontology editing browser," In: *Web Semantics: Science, Services and Agents on the World Wide Web 4.2*, pp. 144-153, 2006.

Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. (2007). Ontology visualization methods a survey. ACM Comput. Surv. 39, 4, 1--43.

Kett, C.; Artale, A. (2008) “Representing and reasoning over a taxonomy of part-whole relations”. In *Applied Ontology 3* (1-2), p. 91-110, 2008.

Malik, Z. H. (2017). “Usability Evaluation of Ontology Engineering Tools”, In: *Computing Conference 2017*, London, UK.

Michael E.; Waterfeld W. (2012). "Overview of the Neon Toolkit," In: *Ontology Engineering in Networked World*, pp. 281-301, 2012.

Mealy, G. H. (1967). Another Look at Data. Proc. of the Joint Computer Conf., pp. 525-534.

Mendonça, F. M (2015). *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. (2016). "*OntoForInfoScience*: a detailed methodology for construction of ontologies and its application in the blood domain". In *Brazilian Journal of Information Science*, v. 10, p. 1.

Munn, K; Smith, B. (2008). "Applied Ontology: An Introduction". Heusenstamm, Germany: Ontos Verlag, 2008.

Musen, M.A. (2015). The Protégé project: A look back and a look forward. AI Matters. Association of Computing Machinery SIG-AI, 1(4), June 2015.

Mylopoulos, J. (1992). Conceptual Modelling and Telo". In P. Loucopoulos, & R. Zicari (Eds.), Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development. New York: Wiley.

Noy, N. F.; McGuinness, D. L (2001). "Ontology Development 101: A Guide to Creating Your First Ontology". In Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.

Poveda-Villálon, M.; Suárez-Figueroa, M.C.; Gómez-Pérez, A. (2010). "A Double Classification of Common Pitfalls in Ontologies". In: OntoQual 2010 - Workshop on Ontology Quality at the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010). *Proceedings of the Workshop on Ontology Quality – OntoQual*, Lisbon, Portugal. Pages: 1- 12.

Poveda-Villalón, M., Suárez-Figueroa, M. C., Gómez-Pérez, A. (2012). "Validating ontologies with oops!" In: *International conference on knowledge engineering and knowledge management* (pp. 267-281). Springer, Berlin, Heidelberg.

Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C. (2004). "Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns". In Proc. of EKAW 2004, pp: 63–81. Springer.

Roussey, C., Corcho, O., & Vilches-Blázquez, L. M. (2009). "A catalogue of OWL ontology antipatterns". In *Proceedings of the fifth international conference on knowledge capture* (pp. 205-206). ACM.

Sales, T. P., & Guizzardi, G. (2015). "Ontological anti-patterns: Empirically uncovered error-prone structures in ontology-driven conceptual models". In *Data & Knowledge Engineering*, 99, 72-10.

Schalley, A.C. (2019). Ontologies and ontological methods in linguistics. Lang Linguist Compass. 2019; 13: e12356. https://doi.org/10.1111/lnc3.12356

Schulz, S.; Kumar, A.; Bittner, T. (2006). "Biomedical ontologies: what part-of is and isn't". In *Journal of Biomedical Informatics*, v. 39, p. 350-361.

Siricharoen, W. V. (2018). “Ontology Editors Approach for Ontology Engineering”. *In: International Conference on Control, Automation and Robotics*, 2018.

Smith, B. Ontology and Information Systems. (2004). Disponível em: <http://www.ontology.buffalo.edu/ontology>. Acesso em: 06 de abril 2020.

Smith, B.; Welty, C. Ontology: Towards a new synthesis. (2001). Disponível em: <http://www.cs.vassar.edu/~weltyc/papers/foisintro.pdf>. Acesso em: 20 dez. 2010.

Suárez-Figueroa, M. C. (2008). “NeOn D 5.4.1. NeOn Methodology for Building Contextualized Ontology Networks”. NeOn project. Available in: http://www.neon-project.org. Acesso em: 06 de abril de 2020.

Vigo, M.; Bail, S.; Jay, C. Stevens, R. (2014). Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. International Journal of Human-Computer Studies Volume 72, Issue 12, 2014, Pages 835-845.

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

Wand, Y.; Storey, V. C.; Weber, R. (1999). An ontological analysis of the relationship construct in conceptual modeling. ACM Transactions on Database Systems, New York, v. 24, n. 4, p. 494-528, 1999.

Warren, P. (2013). Ontology Users' Survey – Summary of Results. The Knowledge Media Institute (KMi). Disponível em: http://kmi.open.ac.uk/publications/pdf/kmi-13-01.pdf Acesso em: 21 de maio 2020.