

KGWE - A customizable editor for OWL ontologies

Wander de Almeida Limeira*

wander.limeira@usp.br

Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, (USP)
São Carlos, SP – Brazil

Dilvan de Abreu Moreira

dilvan@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, (USP)
São Carlos, SP – Brazil

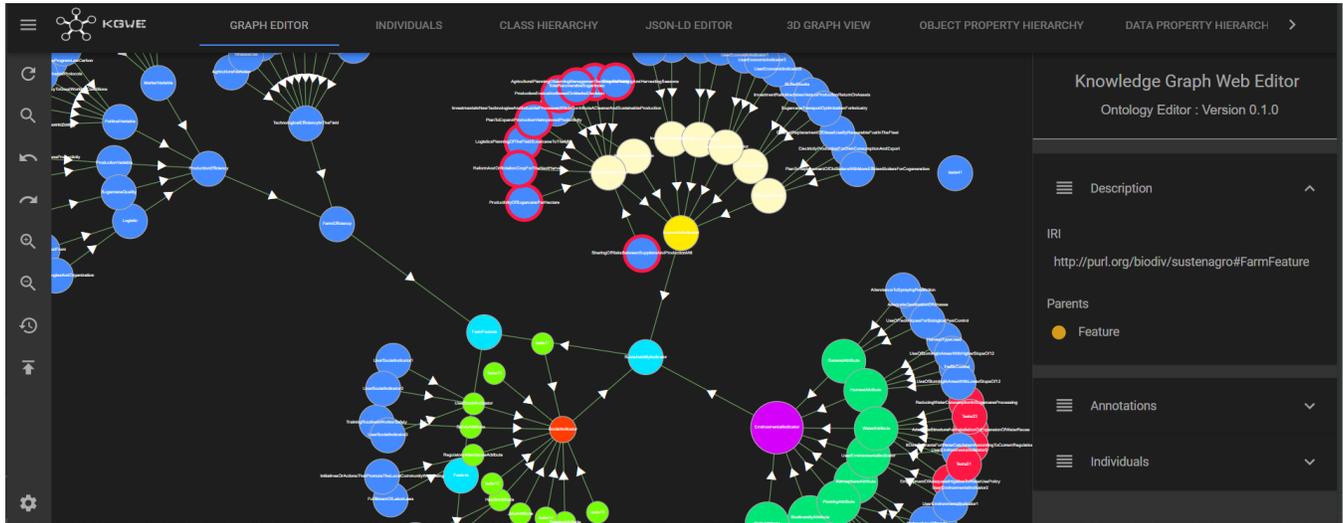


Figure 1: Classes da ontologia representadas com cores e tamanhos personalizadas

ABSTRACT

Regarding ontology editors, it is observed that there are several gaps left by the main tools available on the web for editing and viewing ontologies. The proposal for this work is to provide more efficient and effective editing interfaces, providing a practical tool that improves the visualizing OWL graphs.

KEYWORDS

editor, ontology, OWL

1 INTRODUÇÃO

Ao se deparar com o grande avanço que a web passou na última década, uma coisa chama logo a atenção, o volume crescente de dados, que se expande a ritmos vertiginosos. Quando se fala em Web Semântica, entende-se que as tecnologias e ferramentas envolvidas necessitam de aprimoramentos e inovações constantes para acompanhar tamanha ascendência. O Knowledge Graph Web Editor (KGWE) visa contribuir para o avanço tecnológico das ferramentas de edição e visualização de ontologias, proporcionando um editor simples e ao mesmo tempo eficiente, com uma interface rica em detalhes e altamente customizável.

In: XIX Workshop de Ferramentas e Aplicações (WFA 2020), São Luís, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2020.
© 2020 SBC – Sociedade Brasileira de Computação.
ISSN 2596-1683

Uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada, ou seja, a ontologia descreve os conceitos e também os relacionamentos que existem entre esses conceitos [2]. A OWL (Ontology Web Language) é a linguagem padrão para a criação de ontologias na web. Ela possibilita a criação de ontologias mais expressivas quando comparada, por exemplo, a ontologias em RDF (Resource Description Framework).

Atualmente, a OWL possui suporte em diversos editores. Estudos em visualização de informação tem auxiliado os projetistas dessas ferramentas com diversas técnicas, proporcionando melhorias substanciais nas interfaces de editores de ontologias. A visualização gráfica, um subcampo da visualização de informações, fornece diversas técnicas para visualizações de ontologias. No campo de visualização da informação, Ware[10] descreve os seguintes benefícios em sua utilização:

- Fornece a capacidade de compreender grandes quantidades de dados.
- Permite a percepção de propriedades emergentes que não foram antecipadas.
- Permite que problemas com os próprios dados se tornem imediatamente aparentes.
- Facilita a compreensão dos recursos de grande e pequena escala dos dados.
- Facilita a formação de hipóteses.

Segundo Katifori et al.[3] uma ontologia deve ser exibida de forma que o usuário possa, sem esforço, identificar as seguintes informações:

- Classes: devem ser visualizadas de maneira inteligível;
- Instâncias: representá-las como nós conectados a classes nem sempre é eficaz devido ao seu grande número.
- Taxonomia (relações Isa): são relações de herança entre as classes. O sistema deve, ao menos, fornecer uma representação hierárquica da ontologia;
- Herança múltipla: não são fáceis de representar em combinação com uma representação eficaz da taxonomia;
- Relações: outras relações entre classes, que não pertencem a taxonomia, são difíceis de serem representadas;
- Propriedades: as propriedades associadas a uma entidade são também de muita importância na representação.

Katifori et al. cita ainda que mais pesquisas e avaliações são necessárias para acoplar a visualização e interação de forma eficaz para criar uma ferramenta útil e fácil de ser utilizada.

Um estudo de usabilidade para ferramentas de visualização de ontologias apresenta os seguintes requisitos abaixo [6]. A ferramenta deve:

- Ser consistente;
- Possuir atalhos;
- Retroalimentação;
- Prevenir erros;
- Ser Personalizável.

Em uma revisão sistematizada realizada por Netto and Lima[7], identificou-se que há a necessidade do desenvolvimento de novas pesquisas em visualização de ontologias, sendo os principais campos de investigação:

- Visualização de ontologias complexas;
- Ferramentas para usuários leigos em ontologias;
- Estudos de usabilidade e de requisitos de usuários;
- Contribuição cognitiva para usuários.

2 OBJETIVOS E PROBLEMA DE PESQUISA

O objetivo principal deste trabalho é propor um modelo de notações customizáveis, baseadas em regras, para representação de ontologias em editores gráficos, com o intuito de tornar as interfaces desses editores mais eficientes e também eficazes.

Como objetivos específicos temos:

- A definição de uma arquitetura de notações;
- O editor de ontologias (aqui denominado KGWE);
- A integração do editor em programas que usam ontologias;
- Demonstrar que o uso do editor e das notações permitem que especialistas de domínio obtenham ganhos significativos na visualização e edição de ontologias em OWL.

Como problema de pesquisa podemos realizar o seguinte questionamento: qual a melhor forma de representar uma ontologia em um editor utilizando formatos, dimensões e cores de elementos de forma personalizável?

3 JUSTIFICATIVA E RELEVÂNCIA

Atualmente, existe um projeto de cooperação entre a Universidade de São Paulo - USP e a Embrapa para o desenvolvimento de soluções

computacionais no domínio da agricultura. Um dos trabalhos desenvolvidos nessa parceria é o software Sustenagro, um Sistema de Apoio à Decisão (SAD), baseado em ontologias, que tem como objetivo avaliar a sustentabilidade de sistemas produtivos de cana-de-açúcar na região centro-sul do Brasil.

Ao trabalhar com as ontologias do SustenAgro, a maior necessidade dos especialistas de domínio da Embrapa se refere a simplificação dos conceitos em OWL, presentes nas ontologias, através da sua representação gráfica dentro do editor.

4 TRABALHOS RELACIONADOS

Em um estudo mais aprofundado, é possível perceber que existem diversos editores de ontologias disponíveis para utilização. Porém fica evidente que muitos desses editores possuem diversas limitações ou lacunas. Este trabalho pretende explorar parte dessas limitações e propor novas soluções para problemas já conhecidos. Ao realizar buscas na literatura, para se identificar o estado da arte dos editores de ontologia, buscou-se limitar o foco da pesquisa, considerando os seguintes critérios de classificação e seleção:

- Visualizador e/ou Editor de ontologias;
- Suporte a OWL;
- Disponível na plataforma Web.

Alguns dos trabalhos relacionados serão brevemente descritos nas próximas subseções. Vale ressaltar que as ferramentas destacadas abaixo não possuem mecanismos de personalização de cores, formas e tamanhos de elementos da ontologia, que é o principal diferencial do editor proposto neste trabalho.

4.1 VOWL

A VOWL é uma notação visual para ontologias OWL e define uma linguagem visual para representar os elementos presentes em uma ontologia.

Os componentes do VOWL são formados por um conjunto básico de blocos que consideram formas e cores distintas para representar diferentes aspectos dos elementos da OWL [5].

A WebVOWL é uma aplicação web para visualização de ontologias que faz uso das notações definidas pela VOWL [4]. Ela utiliza um gráfico direcionado por força para representar a ontologia.

4.2 Web Protégé

O Web Protégé, como seu próprio nome já diz, é a versão web da famosa versão do Protégé para desktops. Além de carregar um dos nomes mais importantes no mundo dos editores de ontologias, o Web Protégé vem com a proposta de tornar o desenvolvimento de ontologias mais colaborativo, disponibilizando para isso funções onde um usuário, com uma sessão ativa, pode compartilhar a ontologia com outros usuários e acompanhar as modificações realizadas por uma aba de histórico.

O plano do Web Protégé é fornecer uma plataforma de colaboração que seja facilmente personalizável para diferentes usuários e configurações de projetos [9].

Ao utilizar a ferramenta fica evidente que existem poucas opções de personalização e também poucas funcionalidade de edição para OWL.

5 ESTUDOS DE CASO E PÚBLICO ALVO

Estudos de caso estão sendo realizados junto a Embrapa Meio Ambiente em parceria com a USP e estão em processo de finalização de coleta de dados. Duas ontologias desenvolvidas em estudos anteriores foram utilizadas, sendo elas as ontologias Sustenagro e Nano. A ontologia Sustenagro auxilia na avaliação de sustentabilidade da produção de cana-de-açúcar. Ela representa conceitos como: indicadores, componentes de indicadores, índices, dimensões da sustentabilidade, recomendações e métodos de avaliação [8]. Já a ontologia do sistema de apoio à decisão do Nano foi desenvolvida para que fosse possível avaliar a segurança de plantas geneticamente modificadas em relação a saúde humana e ao meio ambiente por meio do método Risk Assessment Method for Genetically Modified Plants (GMP-RAM) [1].

O público alvo deste trabalho são especialistas de domínio que não trabalham diretamente com ontologias ou conhecem muito pouco OWL. O principal uso do editor se dá por especialistas de domínio envolvidos na utilização de SADS. Este trabalho tem como proposta a validação da hipótese de pesquisa junto aos especialistas de domínio em sustentabilidade. A utilização da ferramenta poderá eventualmente ser expandida também para outros domínios de aplicação em trabalhos futuros.

Não há neste momento nenhuma iniciativa para disponibilização comercial da ferramenta, sendo seu uso puramente acadêmico e também social.

6 KNOWLEDGE GRAPH WEB EDITOR

O editor de ontologias KNOWLEDGE GRAPH WEB EDITOR (KGWE) foi construído com o propósito de atender os requisitos propostos neste trabalho. O KGWE possui diversas interfaces de visualização e edição, proporcionando a seus usuários mecanismos para que se possa criar e modificar ontologias de forma muito eficiente e também eficaz.

A licença da aplicação está sob a GNU General Public License v3.0, cedendo a toda comunidade a liberdade de executar o programa ou mesmo adaptá-lo às suas necessidades.

6.1 Arquitetura da aplicação

A arquitetura do KGWE pode ser separada em diversas camadas. Cada uma dessas camadas foi constituída para ser independente, ou seja, cada camada é executada em um container isolado dentro do servidor.

Em resumo, o KGWE possui três camadas: a camada cliente composta pelo frontend, a camada de serviços composta pela lógica funcional da aplicação disponibilizada por meio das APIs no padrão REST e, por fim, a camada de dados representando a triplestore. Cada uma das três camadas pode rodar na mesma máquina ou em clusters de computadores sem nenhum prejuízo a saúde operacional do sistema. O deploy da aplicação foi realizado em um servidor da própria USP, disponível em: <<http://java.icmc.usp.br:2830/?repository=sustenagro&user=Sample>>. Além disso, uma segunda instância da aplicação foi hospedada por um serviço na nuvem, apenas como redundância.

Testes de uso geral da ferramenta mostraram que a aplicação possui desempenho satisfatório na rotina de carregamento e execução de tarefas diversas. Na próxima sessão, há mais detalhes sobre essas

camadas e uma breve descrição das tecnologias envolvidas em cada uma delas.

6.2 Tecnologias utilizadas

O KGWE é projetado para plataforma Web. Desde o início do projeto havia uma preocupação muito grande quanto ao seu desempenho e usabilidade. Já na fase de concepção do projeto buscou-se a utilização de tecnologias modernas no seu desenvolvimento que garantissem confiabilidade e robustez para toda a aplicação.

6.2.1 Frontend. O frontend foi construído utilizando a biblioteca javascript REACT juntamente com um framework javascript para a linguagem Kotlin. A interface do editor foi toda construída usando conceitos do framework Material Design da Google. A aba principal do editor, o editor gráfico, faz uso da biblioteca javascript D3js, que permite a renderização e manipulação gráfica de elementos da ontologia. A ontologia apresentada nesta aba segue o mecanismo de gráfico direcionado por força, onde o arranjo dos elementos (nós) permite uma visualização muito eficaz da ontologia.

6.2.2 Backend. O backend também foi construído utilizando a linguagem de programação Kotlin. O backend provê diversas APIs, todas elas no padrão REST. A maioria dessas APIs trata de funcionalidades para manipulação de queries em SPARQL. As queries em SPARQL são executadas utilizando o Protocolo definido pela W3C, o SPARQL 1.1 Protocol.

6.2.3 Base de dados. Na base de dados, utilizou-se a triplestore GraphDB para armazenamento de dados da ontologia e também das configurações realizadas pelo especialista de domínio na ferramenta. O GraphDB disponibiliza uma interface web que permite consultas diretamente na base de dados. A interface Web da triplestore também disponibiliza gráficos para que se possa explorar os dados com mais profundidade.

6.3 Componentes do editor

O editor de ontologias KGWE possui diversos componentes, cada qual com um propósito bem definido. Muitos desses componentes possuem funcionalidades semelhantes às encontradas em ferramentas tradicionais de edição de ontologias, como o Protégé. Isso faz com que usuários já habituados com outras ferramentas consigam se adaptar facilmente ao KGWE. Foram desenvolvidos, neste trabalho, conceitos inovadores para interfaces de editores de ontologias em OWL. A seguir serão apresentados os principais componentes do editor e uma breve descrição de suas funcionalidades.

6.3.1 Graph Editor. A aba de edição gráfica possui diversas características inovadoras, a começar pela representação customizável dos nós, onde é possível, por exemplo, alterar a cor de fundo de uma classe ou até mesmo a cor de sua borda, como mostra a Figura 2.

Além das customizações de cores, é possível também realizar a customização do tamanho desses nós de acordo com alguma regra pré-estabelecida. Vale ressaltar que a customização de classes com imagens personalizadas da web também é possível dentro do editor.

6.3.2 Class Hierarchy. A aba de hierarquia de classes apresenta uma visão clássica da ontologia no formato de árvore. Esse tipo de representação é muito útil para se visualizar uma taxonomia. Nessa

- Representações gráficas de classes: apresenta o elemento OWL e o símbolo correspondente no qual a classe está sendo representada;
- Esquema de cores da classe: aqui são apresentadas todas as regras de cores customizadas;
- Esquema de tamanho de classe: ele contém regras de tamanho dos elementos, como, por exemplo, o tamanho padrão das classes genéricas ou tamanho padrão da borda dessas classes.

A Figura 5 mostra como é essa interface.

Graphical representations of classes

OWL	Type	Representation	Description
owl:Class	Circle		Classes in the ontology are represented by circles

Class color scheme

OWL	Color Rule	Precedence	Component Color	Color code	Rule	Value
owl:Class	Background	1		#ff3d00	Class equals =>	VinasseUse
owl:Class	Background	1		#6511ff	Class equals =>	ProductionEfficiency
owl:Class	Background	2		#ff9100	Subclass equals =>	ProductionEfficiency
owl:Class	Background	3		#d500f9	Number of individuals equals:	>= 2 AND <= 2
owl:Class	Background	4		#2979ff	Class equals =>	All Classes
owl:Class	Border	1		#76ff03	Class equals =>	ProductionEfficiency
owl:Class	Border	1		#ff1744	Class equals =>	All Classes
owl:Class	Border	2		#2979ff	Subclass equals =>	ProductionEfficiency
owl:Class	Label	1		#00e676	Class equals =>	All Classes
owl:Class	Relation (link)	1		#ff3d00	Class equals =>	All Classes

Class size scheme

OWL	Size Rule	Precedence	Size	Rule	Value
owl:Class	Radius	1	50px	Class equals =>	ProductionEfficiency
owl:Class	Radius	2	70px	Subclass equals =>	ProductionEfficiency
owl:Class	Radius	3	80px	Number of individuals equals:	>= 2 AND <= 2
owl:Class	Circle Border	1	6px	Class equals =>	ProductionEfficiency
owl:Class	Circle Border	2	8px	Subclass equals =>	ProductionEfficiency

PRINT EXPORT

Figure 5: Conjunto total de notações criadas pelo usuário.

7 AVALIAÇÃO

Com base nas entrevistas realizadas com especialistas e também por meio das respostas obtidas por questionário, foi possível concluir que o editor KGWE e o uso de notações personalizável trouxeram ganhos substanciais aos especialistas de domínio, quando comparado aos métodos tradicionais que estes usuários utilizavam para visualizar ou mesmo editar ontologias.

Foram realizadas entrevistas com os especialistas para coleta de dados sobre o uso do editor e, no final desse processo, os dados foram sumarizados e os resultados evidenciados. Os especialistas foram orientados a realizar algumas tarefas e dizer em voz alta suas impressões. O especialista foi orientado não apenas a refletir sobre o que estava fazendo, mas também verbalizar tudo que estava pensando. Dentre as tarefas executadas no roteiro, havia simples operações, como a criação de uma nova classe, como também tarefas específicas do KGWE, como customizar a cor para um determinado grupo de classes.

No final de todo o processo de avaliação, os especialistas responderam algumas perguntas disponibilizadas por meio de um questionário online.

Quando questionados em uma escala de 0 a 3, onde 0 indicaria **Nada satisfeito** e 3 **Extremamente satisfeito**, pode-se verificar que 100% dos especialistas responderam que ficaram extremamente satisfeitos (3) com o uso da funcionalidade de personalização de cores, sendo que para a funcionalidade de personalização de tamanho, 50% responderam que ficaram muito satisfeitos (2) e a outra metade extremamente satisfeitos (3).

8 CONSIDERAÇÕES FINAIS

Avanços importantes foram alcançados nos últimos meses, seja em relação as novas funcionalidades desenvolvidas no editor, como também na avaliação e resultados obtidos junto aos especialistas.

Este trabalho contribui para o avanço das ferramentas de visualização e edição de ontologias e proporciona aos especialistas de domínio da Embrapa Meio Ambiente melhorias substanciais no processo de modelagem do conhecimento em seus respectivos sistemas de apoio à decisão.

REFERENCES

- [1] Katia Regina Evaristo de Jesus, Anna Cristina Lanna, Fabio Danilo Vieira, Andre Luiz de Abreu, and Denis Ubada de Lima. 2006. A proposed risk assessment method for genetically modified plants. *Applied Biosafety* 11, 3 (2006), 127–137.
- [2] Thomas R. Gruber. 1993. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis.* 5, 2 (June 1993), 199–220. <https://doi.org/10.1006/knac.1993.1008>
- [3] Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, and Eugenia Giannopoulou. 2007. Ontology Visualization Methods: A Survey. *ACM Comput. Surv.* 39, 4, Article 10 (Nov. 2007). <https://doi.org/10.1145/1287620.1287621>
- [4] Steffen Lohmann, Vincent Link, Eduard Marbach, and Stefan Negru. 2015. WebVOWL: Web-based Visualization of Ontologies. In *Proceedings of EKAW 2014 Satellite Events (LNAI)*, Vol. 8982. Springer, 154–158.
- [5] Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. 2014. VOWL 2: User-oriented visualization of ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 266–281.
- [6] Cristiane Mendes Netto and Gercina Ângela Borém de Oliveira Lima. 2017. Requisitos para ferramentas de visualização de ontologias. In *Tendências atuais e perspectivas futuras em organização do conhecimento*. Centro de Estudos Interdisciplinares do Século XX, 351–362.
- [7] Cristiane Mendes Netto and Gercina Ângela Lima. 2017. Visualização de Ontologias: estudos e perspectivas. *Informação & Sociedade* 27, 3 (2017).
- [8] John Freddy Garavito Suarez. 2017. *Ontologias e DSLs na geração de sistemas de apoio à decisão, caso de estudo SustenAgro*. Master's thesis. Universidade de São Paulo.
- [9] Tania Tudorache, Jennifer Vendetti, and Natalya Fridman Noy. 2008. Web-Protege: A Lightweight OWL Ontology Editor for the Web. In *OWLED*, Vol. 432. 2009.
- [10] Colin Ware. 2012. *Information visualization: perception for design*. Elsevier.