# SWRL Editor: Uma ferramenta Web para visualização e edição de regras SWRL

João Paulo Orlando, Adriano Rívolli, Kleberson J. Serique, Dilvan A. Moreira

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
Av. Trabalhador Sancarlense, 400 – São Carlos – SP
orlando@icmc.usp.br, rivolli@icmc.usp.br, serique@icmc.usp.br, dilvan@icmc.usp.br

## ABSTRACT
The Semantic Web is a way to associate explicit meaning to the content of web documents to allow them to be processed directly by machines. To allow this processing, computers need to have access to structured information collections and rule sets to reason about these content. The Semantic Web Rule Language (SWRL) allows the combination of rules and ontology terms, defined using the Web Ontology Language (OWL), to increase the expressiveness of both. However, as rule sets grow, they become difficult to understand and error prone, especially when used and maintained by more than one person. If SWRL is to become a true web standard, it has to be able to handle big rule sets. To find answers to this problem, we first surveyed business rule systems and found the key features and interfaces they used and then, based on our finds, we proposed techniques and tools that use new visual representations to edit rules in a web application. They allow error detection, rule similarity analysis, rule clustering visualization and atom reuse between rules. These tools are implemented in the SWRL Editor, an open source plug-in for Web-Protégé (a web-based ontology editor) that leverages Web-Protégé's collaborative tools to allow groups of users to not only view and edit rules but also comment and discuss about them. We evaluated our solution comparing it to the only three SWRL editor implementations available in the literature and showed that it implements more of the key features present in traditional rule edition systems.

## Categories and Subject Descriptors
H.3.4 [**Information storage and retrieval**]: Systems and Software – Semantic Web.

## General Terms
Algorithms, Management.

## Keywords
SWRL rules; Rule composition; Rule visualization; Data annotation; Semantic Web;

## 1. INTRODUÇÃO
A Web Semântica é uma maneira de explorar a associação de significados explícitos aos conteúdos de documentos presentes na Web, para que esses possam ser processados diretamente ou indiretamente por máquinas [1]. Para possibilitar esse processamento, os computadores necessitam ter acesso a coleções estruturadas de informações (dados e metadados) e a conjuntos de regras de inferência sobre esses conteúdos (que ajudem no processo de dedução automática) para que seja possível o raciocínio automatizado sobre os mesmos [1].

A Web Semântica renovou e aumentou o interesse em sistemas baseados em regras e seu desenvolvimento [2]. A SWRL (*Semantic Web Rule Language*) é a linguagem padrão para regras da Web Semântica, muitas áreas de estudo na computação estão usando SWRL, entre elas podemos citar: Serviços sensíveis ao contexto, gestão de energia em ambientes domésticos, sistemas de e-learning, gerenciamento de SLA (*Service Level Agreement*) para serviços de IPTV (*Internet Protocol Television*), cálculos de redes de co-autoria em redes sociais, sensores em ambientes inteligentes, gerenciamento inteligente de fotos digitais, extração de características (Ex.: de frequência cardíaca) no acompanhamento contínuo de eletrocardiogramas, etc.

Um dos principais problemas no uso de SWRL é que a medida que o conjunto de regras cresce, os desenvolvedores começam a ter problemas no seu gerenciamento. Um conjunto grande de regras se torna difícil de compreender e sujeito a erros, especialmente quando mantido por mais de uma pessoa (colaborativamente). Apesar disso, o SWRL carece de uma ferramenta para a manipulação de regras que seja mais do que um simples editor de texto. Os editores existentes para SWRL, encontramos apenas 3 na literatura, não utilizam as ferramentas e interfaces normalmente encontradas em editores de regras usados em outras áreas [2], notadamente os editores para regras de negócios. Eles carecem de uma quantidade maior de recursos para os seus usuários, como árvores de decisão e agrupamento de regras, facilidades que existem a muito tempo em outras áreas. Essa é a lacuna que este trabalho pretende preencher.

Por essa razão, o objetivo central deste trabalho é desenvolver soluções para implementar um editor de regras SWRL que incorporem as melhores técnicas em Edição/Visualização de outros domínios de regras (ex. Domínio de regras de negócio) para obter, ao final, uma ferramenta mais completa do que as atuais para a Edição/Visualização de regras SWRL.

Para alcançar esse objetivo, o Web-Protégé foi escolhido como base para esse editor por ser a versão Web do Protégé, um editor de ontologias consolidado e de código aberto, que facilita a colaboração entre desenvolvedores de ontologias.

## 2. SWRL
As regras são usadas para inferir novos conhecimentos a partir de bases de conhecimento em OWL [5]. Regras SWRL são divididas em duas partes: antecedente e consequente. Cada regra é uma implicação entre o antecedente e o consequente, que pode ser entendida como: se todas as condições do antecedente são

verdadeiras, então as condições do consequente são executadas. Ambas as partes consistem em uma conjunção de zero ou mais átomos, não permitindo disjunções ou negação.

Cada átomo é composto por um predicado e uma ou mais variáveis (o número de variáveis é determinado pelo tipo do predicado). Os seis tipos de predicados são [6]: *Class*, *Object property*, *Data valued property*, *Data range*, *Same/different*, *Built-in*.

## 3. TRABALHOS RELACIONADOS

Na literatura, apenas são encontradas as seguintes ferramentas para Edição/Visualização de regras SWRL: o SWRLTab, Axiomé e o ACE View. As duas primeiras plug-ins do Protégé 3 e a ultima um plug-in do Protégé 4.

- A ferramenta SWRL Tab é um ambiente de desenvolvimento para regras SWRL que usa a API Protégé-OWL (OWL 1.1). O SWRL Tab tem suporte a edição, execução, sugestões de termos (autocomplete) e identificação de erros sintáticos em regras SWRL.
- O Axiomé usa a mesma API do SWRL Tab. Para o usuário o Axiomé amplia a quantidade de recursos voltados a organização, visualização e edição das regras SWRL [6].
- O ACE View [7] utiliza o *Attempto Controlled English* (ACE) para visualizar e editar ontologias (OWL) e para regras (SWRL) é apenas suportada a visualização. ACE é um subconjunto do Inglês (linguagem natural controlada), responsável por representar o conhecimento de forma inequívoca, ou seja, cada frase representa uma única forma lógica.

Além disso, foram estudados motores de inferência para regras de negócios, pois essas encontram-se em desenvolvimento e uso a muito mais tempo que a própria existência da Web. Isso tornou a investigação dessas ferramentas um passo importante para o levantamento de requisitos da nova ferramenta. A análise destas ferramentas foi conduzida a partir da revisão de [2]. As ferramentas analisadas foram: BRS Rule Track, CLIPS, Drools, Ferramentas ORM (*Object Role Modeling*), Fico Blaze Advisor, IBM ILOG, LibRT Rule Management System (RMS), OpenLexicon, Oracle Business Rules, RuleXpress, SAP NetWeaver Business Rules Management ou QuickRules Editor, Translator (*TRANSlator from LAnguage TO Rules*), Visual Rules.

## 4. SWRL EDITOR

O SWRL Editor foi desenvolvido como um plug-in para o Web-Protégé, a versão web do editor de ontologias Protégé 3 [8]. Protégé 3 é um editor de ontologia gratuito, extensível e de código aberto. Ele foi desenvolvido pelo *Stanford Center of Biomedical Informatics Research* (*Stanford University*). Já por sua vez, o Web-Protégé foi desenvolvido em GWT (*Google Web Toolkit*) para fornecer um ambiente próximo ao desktop. No lado do servidor ele usa a mesma API do Protégé 3 (Protege-OWL).

Uma das características do Web-Protégé é sua arquitetura de plug-ins, pois permite que recursos novos sejam adicionados gradualmente. Com isso, ele é uma plataforma flexível, mas ao mesmo tempo sólida para o desenvolvimento de tecnologias da Web Semântica. Como um plug-in, o SWRL Editor implementa uma *tab* e uma por *portlet* (componentes para adição de novos plug-ins no Web-Protégé) e através de RPCs comunica com o servidor.

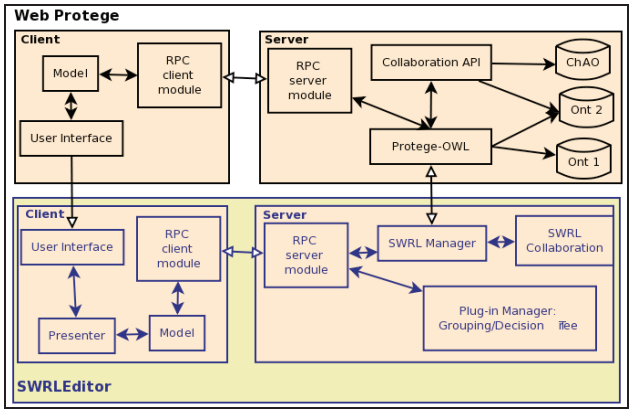


**Figura 1.Integração entre o SWRL Editor e o Web-Protégé.**

A Figura 1 mostra a integração das arquiteturas do SWRL Editor e Web-Protégé. No lado do Web-Protégé, a Protégé-OWL permite a manipulação das ontologias e a *Collaboration API* apóia os serviços de colaboração, tais como, anotação e controle de alterações em termo da ontologia. No lado do Cliente do Web-Protégé, existe uma copia parcial da ontologia (*Model*) e um módulo de RPC (para comunicar com o servidor).

Já a arquitetura do SWRL Editor está dividida em Servidor (Seção 4.1) e Cliente (Seção 4.2). O Servidor é responsável por modificar e armazenar os arquivos da ontologia. Já no Cliente estão as interfaces gráficas para: representação, edição e busca de regras, entre outras funções.

### 4.1 Servidor

O Servidor centraliza as operações com as regras, por exemplo, ontologias e regras são carregadas de seus arquivos apenas uma vez e ficam disponíveis para todos os clientes através de chamadas RPC. Além disso, a centralização das operações é um ponto chave para que a ferramenta possa funcionar colaborativamente, já que o servidor atua como nó central de distribuição das alterações feitas por cada usuário.

#### *4.1.1 Gerenciador*

O gerenciador (*SWRL Manager* da Figura 1) foi criado para intermediar todas as consultas ou modificações no conjunto de regras da ontologia com a API Protégé-OWL. Ele converte os elementos da ontologia, passando do formato da API Protégé-OWL para uma representação do SWRL Editor e na edição de uma regra fazem o processo inverso. Por esse fato, o gerenciador torna-se importante, dado que a ferramenta fica menos dependente arquitetura do Web-Protégé. Esse requisito foi considerado importante, pois a versão 4 do Protégé utiliza novas APIs não compatíveis com as utilizadas no Web-Protégé. Para facilitar uma futura migração para essas novas APIs, foi estabelecida uma representação interna do SWRL Editor para regras.

#### *4.1.2 Gerenciador de atualizações*

O Web-Protégé, por tratar-se de uma ferramenta de edição colaborativa de ontologias, possui um gerenciador de mudanças na ontologia (*Collaboration API*) [8]. Porém, esse gerenciador de atualizações não controla alterações nas regras SWRL. Então, a partir da metodologia empregada na *Collaboration API*, foi criado um gerenciador de atualizações para as regras.

Nesse novo gerenciador (*SWRL Collaboration* da Figura 1), a cada nova alteração nas regras é incrementado um número de versão contido no Servidor. Também é mantida uma lista de alterações, contendo o tipo de alteração e o que foi modificado. Já no cliente é mantido um conjunto de regras com seu número de versão. Em um intervalo definido nas configurações do Web-Protégé (por padrão 5 segundos), o cliente faz uma chamada RPC em que envia o número da versão local das regras para o servidor. Já do servidor, é retornada uma lista de alterações, que vão desde a versão do cliente até a versão do servidor. Quando o cliente recebe as alterações, ele atualiza o seu conjunto de regras e automaticamente os recursos de interfaces as mostram.

Só o gerenciador de atualizações modifica o conjunto de regras do cliente. As operações que modificam os conjuntos de regras (inserir, editar e excluir) enviam suas alterações para o servidor e, caso essas alterações sejam aceitas e implementadas no servidor, é retornado ao cliente o aviso de sucesso na operação e esse força o gerenciador de atualizações a buscar imediatamente pelas novas alterações.

### *4.1.3 Sistemas de carregamento automático de algoritmos*

Um ponto bastante discutido neste trabalho foi a questão de quanto um algoritmo de visualização [15] é importante para várias ontologias ou vários usuários. Mais de um algoritmo foi desenvolvido para criar essas visualizações, porém mesmo assim não é possível garantir que eles sempre sejam úteis para todos os usuários. Por esse motivo, no SWRL Editor foi implementado um sistema para carregamento automático de algoritmos para criação de agrupamentos [4] e árvores de decisão [9].

O sistema de carregamento automático de algoritmos tornou-se uma ótima estratégia para usuários mais avançados que tenham interesse em desenvolver seus próprios algoritmos. Ele facilita essa operação já que para inserir um novo algoritmo é apenas necessário gerar um arquivo JAR com as implementações de interfaces Java e o colocá-lo na pasta em que está o servidor do SWRL Editor.

## 4.2 Cliente

Já no Cliente, as principais funcionalidades implementadas foram: utilização de rdfs:Label na visualização e edição de regras SWRL (Seção 4.2.1) e melhorias na composição de regras (Seção 4.2.2).

### *4.2.1 rdfs:Label*

O SWRL Editor permite aos usuários trabalharem com os rdfs:Labels. O rdfs:Label é uma descrição para o termo da ontologia (representado pelo seu identificador único rdf:ID). Os rdfs:Labels podem tornar mais fácil o entendimento por não serem apenas um ID. Usando a ontologia para categorizar os fenótipos de autismo [3], na Tabela 1 é mostrado alguns rdf:ID de termos da ontologia com o seu respectivo rdfs:Label. Fica evidente que os rdfs:Labels podem tornar mais fácil o entendimento e a edição das regras.

**Tabela 1. rdf:ID versus rdfs:Label**

| rdf:ID | rdfs:Label |
|---|---|
| ados1:AUTISMC000004 | Autism Diagnostic Observation Schedule |
| ados4:AUTISMC000001 | Autism Diagnostic Observation Schedule |
| Autism-core:AUTISMC100000 | Phenotype Record |

Como rdfs:Labels não são únicos, podem ocorrer redundância na edição, pois um mesmo rdfs:Label (descrição) pode estar em mais de um termo da ontologia. Um exemplo está na Tabela 1 em que os termos `ados1:AUTISMC000004` e `ados4:AUTISMC000001` possuem uma mesma descrição. A solução para isso está apresentada na próxima seção.

### *4.2.2 Composição*

O processo de criação de regras disponibiliza dois modos de trabalho [4]:

- Editor – Utiliza a representação hierárquica para apresentar e organizar os átomos da regra e um formulário para editar;
- SWRL – Editor SWRL com *Highlight*, no qual o usuário escreve a regra utilizando a sintaxe SWRL;

As contribuições deste trabalho para a composição foram a sugestão de termos (autocompletar) e o uso dos rdfs:Labels na edição das regras. A primeira funcionalidade consiste em sugerir termos à medida que o nome de algum elemento da ontologia é digitado. Como ontologias podem ser muito grandes, por padrão, cada vez que o usuário acrescenta mais uma letra o autocompletar sugere no máximo 8 termos da ontologia (a medida que o usuário digita mais letras um desses 8 termos vai ser o que ele procura).

Na Figura 2 (A) é mostrada a sugestão de termos por meio do autocompletar. O usuário digitou "Au" e o autocompletar sugeriu 4 termos. Nesse exemplo, os rdfs:Labels (descrições do identificadores únicos da ontologia) estão sendo usados como nomes das classes OWL. Além de rdfs:Labels terem nomes mais intuitivos, geralmente com poucas letras já se consegue uma lista de poucos termos.

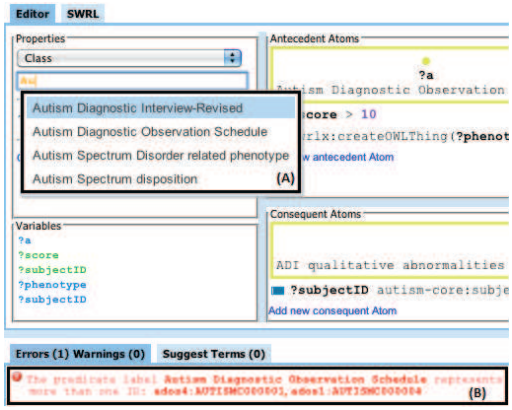


**Figura 2. Composição – (A) Autocompletar; (B) Mensagem de erro de redundância de rdfs:Labels.**

Foi também implementado o tratamento de erros de redundância que os rdfs:Labels podem gerar. Na Figura 2 – (B) é mostrada a seguinte mensagem de erro: "`The predicate label` **`Autism Diagnostic Observation Schedule`** `represents more than one ID:` **`ados4:AUTISMC000001, ados1:AUTISMC000004`**". O usuário poderá então usar um dos dois rdf:IDs para representar o elemento e assim evitar o erro.

# 5. COMPARAÇÃO DE FERRAMENTAS

Nesta seção, é apresentada uma comparação entre recursos/interfaces das ferramentas de manipulação de SWRL que são encontradas na literatura. São comparadas ao SWRL Editor as seguintes ferramentas: SWRL Tab, Axiomé e ACE View. Uma

vantagem sobre as demais é o fato de que seus recursos estão disponíveis na Web e, por essa razão, podem beneficiar-se dos recursos de colaboração que a Web trás e, mais especificamente, dos recursos que o Web-Protégé trás para colaboração.

Para fazer a comparação, utilizaremos os principais recursos/interfaces encontrados em sistemas de regras de negócios e regras SWRL [2]. A Tabela 2 mostra a comparação entre as quatros ferramentas (para o SWRL Editor são incluídos os recursos desenvolvidos em [4] e [9]):

**Tabela 2. Comparação das características entre as principais ferramentas para edição de regras SWRL**

| Recurso/ Interface | SWRL Editor | SWRL Tab | Axiomé | ACE View |
|---|---|---|---|---|
| Plataforma | Web | Desktop | Desktop | Desktop |
| Tabela de decisão | | | | |
| Árvore de decisão | ✓ | | | |
| Diagrama gráfico | | | ✓ | |
| Editor de texto | ✓ | ✓ | ✓ | |
| Editor de texto com *Highlight* | ✓ | | | |
| Formulário/ *Template* | ✓ | | ✓ | |
| Linguagem natural | Parcial | | Parcial | ✓ |
| Representação visual | ✓ | | ✓ | |
| Agrupamento | ✓ | | ✓ | |
| Sugestão de termos | ✓ | ✓ | | |
| Mecanismos de busca | ✓ | ✓ | ✓ | ✓ |
| Customização | ✓ | | | |
| Adição de novos algoritmos | ✓ | | | |
| Edição de Regras com rdfs:Label | ✓ | | | |
| Detecção de Erros | ✓ | ✓ | ✓ | |
| Execução das regras | ✓ | ✓ | ✓ | |

# 6. CONCLUSÃO

A linguagem SWRL é muito útil para desenvolvedores de ontologias, pois suas regras podem fazer inferências de novos conhecimentos sobre indivíduos de uma ontologia (OWL). Um dos principais problemas no uso de SWRL é que grandes conjuntos de regras são mantidos de forma colaborativa e, à medida que crescem, os desenvolvedores começam a ter problemas no seu gerenciamento. Para auxiliar no desenvolvimento desses conjuntos de regras, foi desenvolvido um Editor/Visualizador colaborativo de regras SWRL. Para isso foram incorporadas as melhores técnica em Edição/Visualização de outros domínios de regras (ex. Regras de negócio). Ampliando o conjunto de recursos desenvolvidos em [4] obteve-se o SWRL Editor, o Editor/Visualizador de código aberto mais completo para regras SWRL disponível no momento.

Após seu desenvolvimento, foi feita uma comparação com as outras três ferramentas de Edição/Visualização de SWRL disponíveis na literatura. Nessa comparação foi possível constatar que o SWRL Editor tem vantagem sobre elas, pois ele possui a maioria dos recursos citados em [2]. Além disso, até mesmo recursos que já estão em outras ferramentas para SWRL foram melhorados.

Outra contribuição foi mostrar que é possível fazer uma ferramenta Web para a edição de regras SWRL que tem os recursos de diversas ferramentas desktop. Isso só é possível agora devido a novas tecnologias da Internet, como HTML 5. Além disso, por estar na Web, foi adicionado o recurso de colaboração para os desenvolvedores de regras SWRL.

Como trabalhos futuros, podem ser citados: Detecção de erros semânticos; Detecção de relevância e relações semânticas entre as Regras e Exploração de técnicas que visualizem a localização das regras dentro do contexto da ontologia.

# 7. AGRADECIMENTOS

Esse trabalho contou com o financiamento do CNPq.

# 8. REFERÊNCIAS

[1] Berners-Lee, T., Fischetti, M. 2008. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web. HarperSanFrancisco, ISBN: 978-0062515872.

[2] Zacharias, V. 2008. Development and verification of rule based systems – a survey of developers. Rule Representation, Interchange and Reasoning on the Web: International Symposium, Orlando, Florida, USA, (5321) 6-16. DOI=10.1007/978-3-540-88808-6_4.

[3] Hassanpour, S., O'connor, M. J., Das, A. K. 2011. Evaluation of Semantic-Based Information Retrieval Methods in the Autism Phenotype Domain. AMIA Annual Symposium, Washington, DC, 569–577.

[4] Silva, A. R. 2012. Aprimorando a visualização e composição de regras SWRL na Web. Dissertação - Instituto de Ciências Matemáticas e de Computação (USP), acesso julho 2012.

[5] O'Connor, M. J., Knublauch, H., Tu, S., Grosof, B., Dean, M., Grosso, W., Musen, M. 2005. Supporting Rule System Interoperability on the Semantic Web with SWRL. Fourth International Semantic Web Conference (ISWC2005), Galway, Ireland, (2005). Disponível em: http://bmir.stanford.edu/file_asset/index.php/1157/BMIR-2005-1080.pdf, acesso em Maio de 2012.

[6] Hassanpour, S., O'Connor, M. J., Das, A. K. 2009. Exploration of SWRL Rule Bases through Visualization, Paraphrasing, and Categorization of Rules. Proceedings of the 2009 International Symposium on Rule Interchange and Applications (RuleML 2009), Las Vegas, Nevada, 246–261. DOI=10.1007/978-3-642-04985-9_23.

[7] Kaljurand, K. 2008. ACE View – an ontology and rule editor based on Attempto Controlled English. Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions (OWLED 2008), (432) 1-12. DOI=10.5167/uzh-8822.

[8] Tudorache, T., Vendetti, J. e Noy, N. F. 2008. Web-Protégé: A Lightweight OWL Ontology Editor for the Web. OWLED 2008, Karlsruhe, Germany.

[9] Orlando, J. P. 2012. Usando aplicações ricas para internet na criação de um ambiente para visualização e edição de regras SWRL. Dissertação - Instituto de Ciências Matemáticas e de Computação (USP), acesso julho 2012.