

# Construindo Axiomas e Avaliando Ontologias em ODEd

Vítor Estêvão Silva Souza, Ricardo de Almeida Falbo

Departamento de Informática da Universidade Federal do Espírito Santo

Av. Fernando Ferrari, CEP 29060-900, Vitória – ES – Brasil

vitorsouzabr@yahoo.com.br, falbo@inf.ufes.br

**Abstract.** *Ontologies have been very used in Software Engineering. However, its development is not an easy task. In the context of ODE (Ontology-based software Development Environment), an ontology editor, ODEd, was developed to support ontology construction. Nevertheless, ODEd does not deals adequately with axiom definition. This paper presents AxE, an axiom editor that complements ODEd, allowing axiom building and ontology evaluation using an inference machine.*

**Resumo.** *Ontologias têm sido bastante utilizadas na Engenharia de Software, porém sua construção não é tarefa simples. No contexto do ambiente de desenvolvimento de software ODE, foi desenvolvido ODEd, um editor de ontologias. Contudo, ODEd trabalha de forma limitada a definição de axiomas. Esse artigo apresenta AxE, um editor de axiomas que complementa ODEd, permitindo a construção de axiomas e a avaliação de ontologias através do uso de uma máquina de inferência integrada.*

## 1 – Introdução

Ontologias são bastante úteis para descrever e organizar conhecimento de domínio. Porém, a construção de ontologias não é uma tarefa simples. Envolve a especificação de conceitos e relações que existem em um domínio de interesse, além de suas definições, propriedades e restrições, descritas na forma de axiomas [1].

O processo de construção de ontologias começa com a identificação do propósito e especificação de sua competência. Segue para a captura da ontologia e termina com a formalização da mesma. Em paralelo, encontram-se atividades de avaliação, documentação e integração com ontologias existentes [1].

O ambiente ODE (*Ontology-based software Development Environment*) [2] possui um editor de ontologias, ODEd (*ODE's Ontology Editor*) [3], que suporta o desenvolvimento de ontologias usando representações gráficas, além de promover a geração automática de alguns tipos de axiomas, a derivação de *frameworks* de objetos a partir das ontologias e a instanciação de ontologias. ODEd foi construído para apoiar uma abordagem de engenharia de domínio baseada em ontologias em ODE.

Entretanto, ODEd trabalha com axiomas de uma forma restrita. Foi definido um conjunto de propriedades de relações, que inclui as seguintes: anti-reflexibilidade, anti-simetria, atomicidade, disjunção, exclusividade, reflexibilidade, transitividade e simetria. O engenheiro de ontologias pode, então, indicar quais dessas propriedades são válidas para uma relação da ontologia, definindo, assim, uma teoria associada a essa relação. Deste modo, os axiomas definidos por essas propriedades são associados às relações da ontologia. Por exemplo, se uma dada auto-relação  $r$  é definida como sendo transitiva, o seguinte axioma é implicitamente associado a essa auto-relação:  $(\forall x1,x2,x3) (r(x1,x2) \wedge r(x2,x3) \rightarrow r(x1,x3))$ . Ao derivar o *framework* de objetos a partir dessa ontologia, as propriedades são mapeadas em métodos que permitem checar se as condições estão sendo cumpridas. Essa abordagem, no entanto, apresenta as seguintes limitações:

- O engenheiro de ontologias não pode criar seus próprios axiomas, estando limitado aos axiomas definidos pelas propriedades;
- Sem uma representação explícita dos axiomas e sem uma maneira de permitir que o engenheiro de ontologias faça consultas (*queries*) à ontologia, não é possível apoiar a atividade de avaliação de uma ontologia.

De fato, embora a maioria das fases do processo de desenvolvimento de ontologias seja suportada por ODEd, atividades importantes, como a formalização e a avaliação de ontologias, não foram completamente tratadas. Assim, para permitir escrever quaisquer axiomas na ontologia (e não apenas aqueles capturados na forma de propriedades de relações) e avaliar essa ontologia, verificando se ela atende às questões de competência colocadas como requisito, é fundamental adicionar ao editor capacidades para tratar adequadamente axiomas.

Este artigo apresenta AxE (*Axiom Editor*), um editor de axiomas construído para complementar ODEd, que permite a escrita de axiomas e a avaliação de ontologias. A seção 2 apresenta sucintamente ODEd e discute a infra-estrutura utilizada no desenvolvimento de AxE. Na seção 3, o construtor de axiomas é descrito. A seção 4 apresenta a funcionalidade de consulta e avaliação de ontologias. Por fim, a seção 5 traz as considerações finais do trabalho.

## 2 – ODEd e a Infra-Estrutura de Apoio à Edição de Axiomas em AxE

ODEd é uma ferramenta de apoio a uma abordagem de engenharia de domínio baseada em ontologias. Sendo assim, ODEd apóia a análise de domínio usando ontologias, oferecendo funcionalidades para apoiar a definição de conceitos e relações com o uso de representações gráficas e promovendo a geração automática de alguns tipos de axiomas, e o projeto de domínio, gerando *frameworks* de objetos a partir das ontologias criadas. Além disso, para permitir a investigação de domínio, ODEd gera automaticamente tutoriais sobre o domínio baseados nas ontologias desenvolvidas, usando XML, XSL e HTML [3].

No que se refere ao processo de desenvolvimento de ontologias, ODEd apóia as seguintes atividades [3]:

- **Especificação de Requisitos:** O engenheiro de ontologias pode definir as questões de competência da ontologia, isto é, questões que a ontologia tem de ser capaz de responder. Estas questões são escritas em linguagem natural.
- **Captura da Ontologia:** ODEd permite que sejam construídos modelos gráficos das ontologias, usando LINGO (LINGuagem Gráfica para escrever Ontologias) [1] ou UML. Além disso, certos tipos de axiomas, descritos por propriedades de relações, podem ser definidos.
- **Documentação:** Através dos tutoriais gerados por ODEd, é possível navegar pelos conceitos e relações da ontologia.
- **Integração com Ontologias Existentes:** ODEd suporta a integração de ontologias de um modo muito simples: através da importação de conceitos de ontologias existentes para a ontologia corrente.

Pode-se perceber que ODEd não trata adequadamente do apoio às atividades de formalização e avaliação. De fato, ODEd apóia parcialmente a formalização, uma vez que permite a captura de certos tipos de axiomas, e não oferece nenhum suporte à avaliação de ontologias. A meta principal de AxE é, portanto, permitir que um engenheiro de ontologias, usando ODEd, possa escrever quaisquer axiomas e fazer consultas à ontologia, de modo a avaliar se ela satisfaz às questões de competência colocadas como requisito. Para isso, é imprescindível a utilização de uma linguagem para a descrição de ontologias que possua um mecanismo de inferência associado.

Em AxE, foi adotada inicialmente a linguagem DAML+OIL [4] para descrever ontologias. Esta escolha deveu-se ao fato da mesma estar sendo considerada para se tornar um padrão W3C para descrição de ontologias. Além disso, para esta linguagem, há disponível o mecanismo de inferência JTP (*Java Theorem Prover*) [5]. No entanto, como no estágio atual DAML+OIL não possui muito poder de expressão, a escrita de axiomas ainda é limitada nesta linguagem. Tendo em vista esta limitação e considerando que o mecanismo de inferência JTP também trata a linguagem KIF [6], esta linguagem foi também adotada, visando ampliar o poder de expressão na escrita de axiomas. Desta forma, as ontologias produzidas em ODEd são traduzidas para o formato DAML+OIL/KIF e podem ser submetidas a avaliação, usando JTP. São características importantes de JTP para AxE:

- JTP lê ontologias escritas em DAML+OIL e aceita axiomas escritos em KIF, permitindo fazer inferências sobre as ontologias descritas nessas linguagens;
- É modular, orientado a objetos e escrito em Java, linguagem de programação utilizada no projeto ODE;
- É gratuito e possui o código-fonte aberto.

Em ODEd, as ontologias são armazenadas como classes persistentes no repositório central de ODE. Ao adotar JTP, um mecanismo de inferência externo, faz-se necessário converter a ontologia de seu formato interno em ODE para as linguagens suportadas por JTP. A figura 1 ilustra como se dá o funcionamento de AxE e as novas funcionalidades que ele acrescenta a ODEd. AxE adiciona a ODEd duas funcionalidade principais: permitir que o engenheiro de conhecimento escreva seus próprios axiomas (seção 3) e permitir que ele avalie uma ontologia através perguntas enviadas a um mecanismo de inferência (seção 4).

### 3 – Editando Axiomas com AxE

Para adicionar um axioma à ontologia, o engenheiro de ontologias pode utilizar o Construtor de Axiomas, mostrado na figura 2. No exemplo mostrado, utilizou-se a ontologia de processo de software, descrita em [1]. Nesta ontologia, o seguinte axioma deve ser observado: se uma atividade X é sub-atividade de uma atividade Y, e se Y é pré-atividade de uma atividade Z, segue que X é pré-atividade de Z. Esse axioma não é contemplado por nenhuma das propriedades pré-definidas em ODEd e precisa, portanto, ser construído pelo engenheiro de ontologias.

A interface mostrada na figura 2 foi criada com o objetivo de atender tanto a engenheiros de ontologias mais experientes quanto aos novatos, sobretudo em KIF. Na coluna da esquerda são listados os conceitos, relações e instâncias (de conceitos) da ontologia com a qual o usuário está trabalhando, além de uma lista de variáveis por ele criadas. Através dos botões correspondentes, pode-se incluir quaisquer destes elementos em um axioma. Na parte inferior estão os operadores que podem ser utilizados, vindos da linguagem KIF, além da propriedade `rdf:type`, vinda do RDF(S), que modela relações de instanciação de conceitos [Broekstra et al. 2001].

O axioma deve ser escrito em KIF, que é uma linguagem pré-fixada, o que faz dela uma linguagem difícil para aqueles que ainda não estão habituados com esse tipo de escrita. Para contemplar essa classe de usuários, AxE permite que se escreva axiomas tanto de forma infixa quanto prefixa. O exemplo da figura 2 mostra o axioma escrito em KIF infixo. AxE é responsável por converter o axioma para o formato prefixo antes de enviá-lo para o JTP. Finalmente, para apoiar o entendimento do axioma, é possível fornecer uma descrição em linguagem natural.

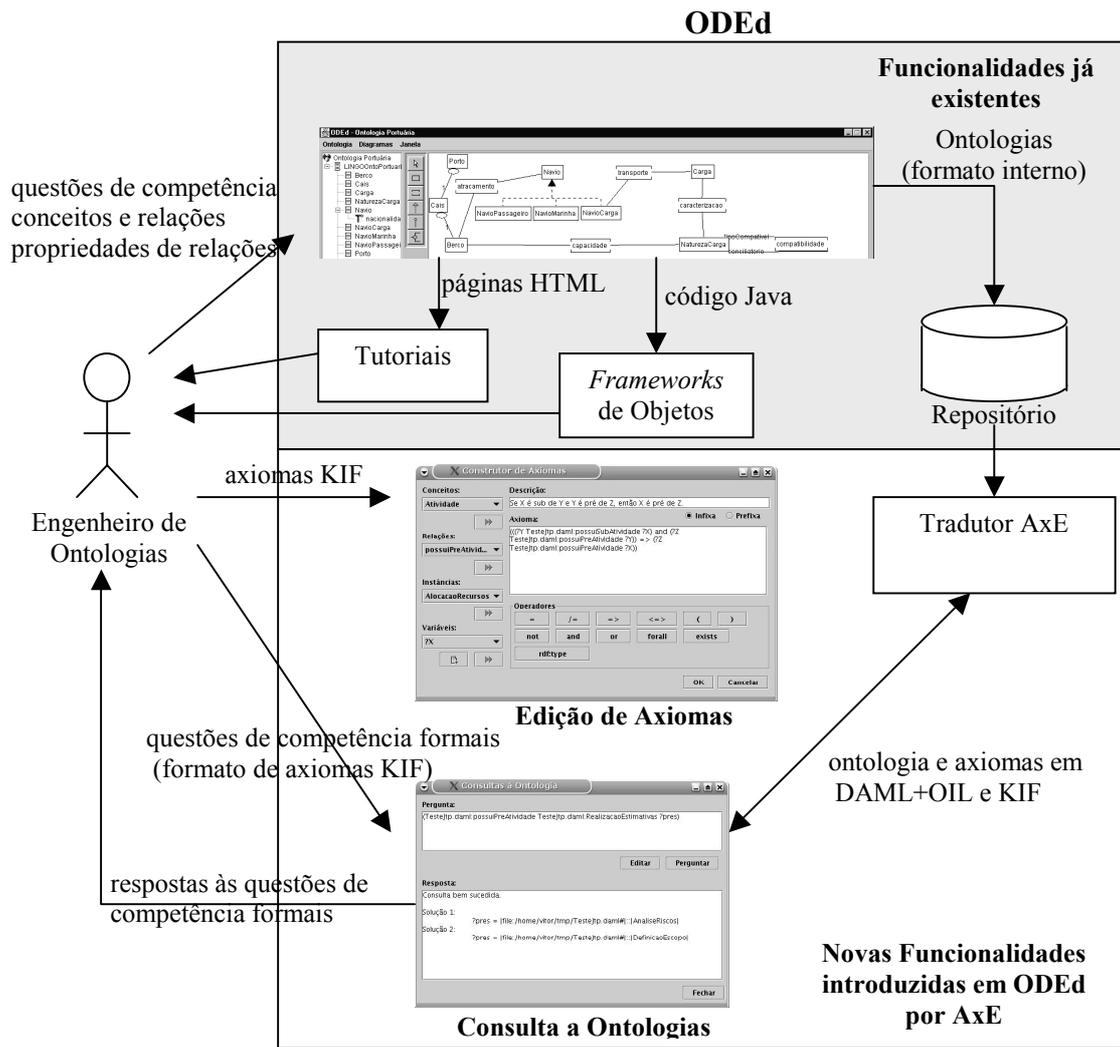


Figura 1 – Estendendo ODEd com AxE.

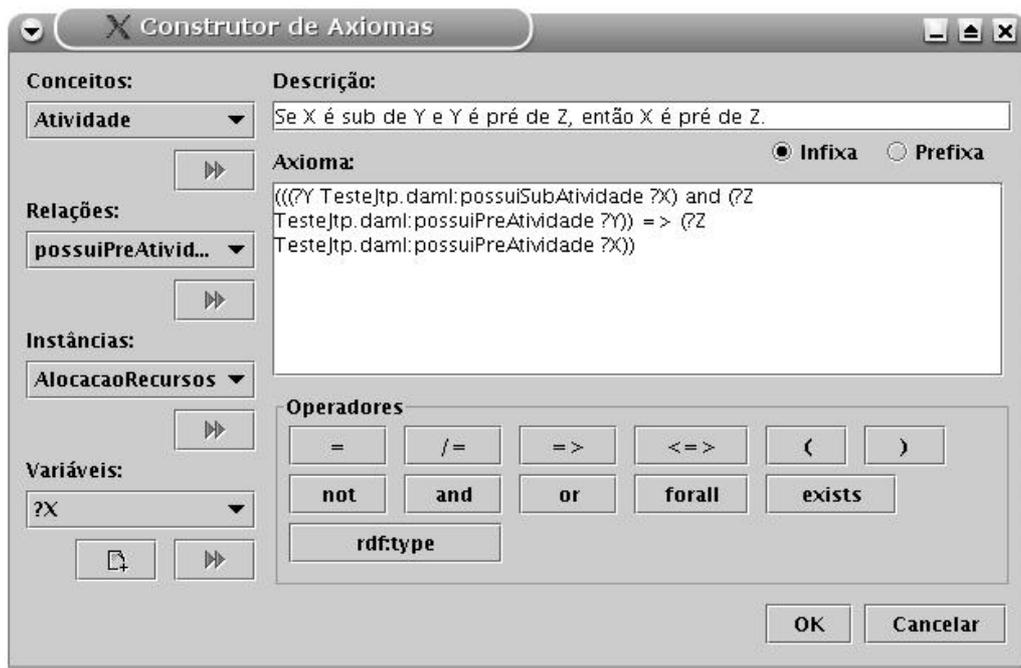


Figura 2 – Editor de Axiomas

## 4 – Avaliando Ontologias com AxE

A segunda funcionalidade que AxE adiciona a ODEd é a possibilidade de efetuar consultas à ontologia através do mecanismo de inferência. Isso dá ao engenheiro de ontologias a capacidade de checar a consistência de sua ontologia, verificando, por exemplo, se ela responde às questões de competência relacionadas.

Para que isso seja possível, primeiro, é necessário traduzir a ontologia para as linguagens suportadas pelo mecanismo de inferência. Nesse sentido, AxE converte a ontologia que está gravada como objetos persistentes para um arquivo no formato DAML+OIL e axiomas KIF. Vale ressaltar que são escritas em KIF somente as informações que não podem ser expressas em DAML+OIL.

Em seguida, o mecanismo de inferência é alimentado com o arquivo DAML+OIL e os axiomas KIF (tanto os gerados automaticamente quanto os escritos pelo engenheiro de ontologias utilizando o Editor de Axiomas). Após esse processo, o mecanismo de inferência está pronto para responder às perguntas do usuário. Para efetuar suas perguntas, o engenheiro de conhecimento deve utilizar a interface mostrada na figura 3. No campo superior, o engenheiro de ontologias deve escrever a pergunta que deseja verificar, em formato KIF (prefixo). Caso queira usar o construtor de axiomas, mostrado na figura 2, para apoiar a escrita da questão, basta clicar no botão “Editar”. Ao terminar de formular sua consulta, o usuário deve clicar no botão “Perguntar” e o mecanismo de inferência responde à consulta mostrando o resultado na parte inferior da tela.

O exemplo da figura 3 trata da verificação da seguinte questão de competência da ontologia de processo de software [1]: “Quais as pré-atividades de uma atividade?”. Visando verificar essa questão de competência, o engenheiro de ontologias pode perguntar quais as pré-atividades de uma atividade específica, tal como a instância do conceito atividade “Realização de Estimativas”. AxE responde que são duas as atividades: “Análise de Riscos” e “Definição de Escopo”.

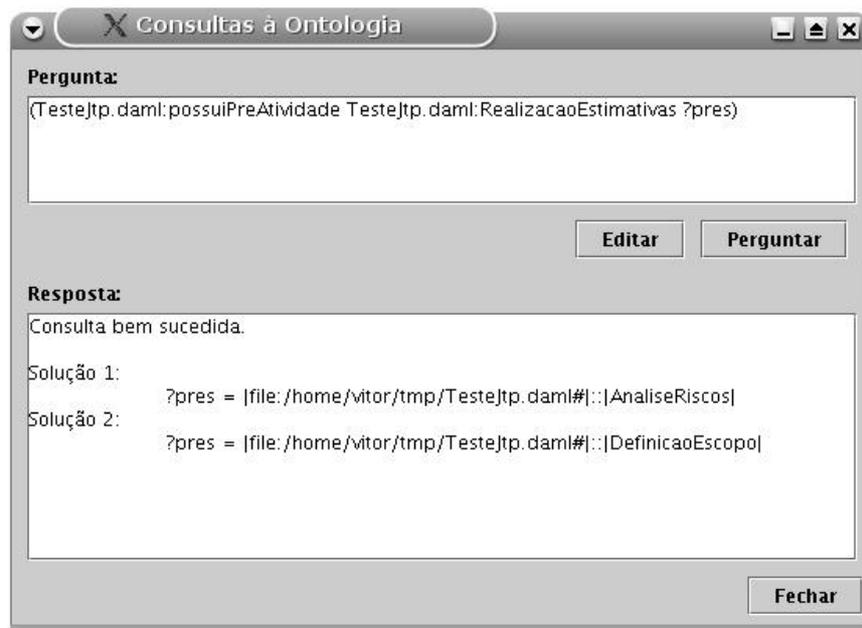


Figura 3 – Avaliando Ontologias

Desta forma, através de consultas à ontologia, o engenheiro de ontologias pode avaliar se a ontologia responde às questões de competência e, portanto, se satisfaz os requisitos para ela definidos.

## 5 – Considerações Finais

Esse artigo apresentou AxE, um editor de axiomas que permite a edição de axiomas e a avaliação de ontologias através do uso de uma máquina de inferência integrada. AxE permite que o engenheiro de ontologias defina seus próprios axiomas e faça consultas à ontologia para avaliá-la. Desta forma, AxE adiciona ao editor de ontologias ODEd funcionalidades necessárias para apoiar com maior eficácia algumas atividades do processo de desenvolvimento de ontologias, principalmente a formalização e a avaliação de ontologias.

Entretanto, AxE possui algumas limitações. AxE está dependente do JTP e, como tal, não permite que sejam usadas conjuntamente regras de encadeamento progressivo e regressivo (*forward / backward chaining rules*). Assim, os axiomas têm de ser montados sempre respeitando essa restrição. Além disso, o processo de carga das informações no JTP é relativamente lento.

A interface para construção de axiomas poderia ser mais amigável. De fato, idealmente, ela deveria ser integrada ao construtor de diagramas de ODEd. Além disso, as respostas às consultas feitas ao JTP poderiam também ser apresentadas em um formato mais amigável, e não somente uma listagem em formato texto.

## Agradecimentos

Os autores agradecem ao CNPq pelo apoio financeiro dado a este trabalho.

## Referências

- [1] Falbo, R.A., *Integração de Conhecimento em um Ambiente de Desenvolvimento de Software*, Tese de Doutorado, COPPE/UFRJ, 1998.
- [2] Bertollo, G.; Ruy, F.B.; Mian, P.G.; Pezzin, J.; Schwambach, M.; Natali, A.C.C.; Falbo, R.A., “ODE – Um Ambiente de Desenvolvimento de Software Baseado em Ontologias”. Anais do XVI Simpósio Brasileiro de Engenharia de Software, Caderno de Ferramentas, Gramado, Outubro 2002.
- [3] Mian, P.G., Falbo, R.A., “Supporting Ontology Development with ODEd”, Proceedings of the 2nd Ibero American Symposium on Software Engineering and Knowledge Engineering. Salvador, Brazil, October 2002.
- [4] Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider P.F., and Stein, L.A., "DAML+OIL (March 2001) Reference Description," December. 2001. Available: <http://www.w3.org/TR/daml+oil-reference>.
- [5] Gleb F., "A General Interface for Interaction of Special-Purpose Reasoners within a Modular Reasoning System", In: Question Answering Systems. AAAI Fall Symposium, pp. 57-62, 1999.
- [6] Genesereth, M.E. and Fikes, R.. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-921, Computer Science Department, Stanford University, 1992.
- [7] Broekstra, J., Klein, M., Decker, S., Fensel, D., van Harmelen, F. and Horrocks, I. Enabling knowledge representation on the web by extending rdf schema. In Proceedings of the tenth World Wide Web conference WWW'10, Hong Kong, May 2001.