

The logo for 'OMNIA EXATAS' features the word 'OMNIA' in large, bold, yellow 3D-style letters. The word 'EXATAS' is written vertically in smaller, black, uppercase letters on a yellow rectangular background that is positioned between the 'N' and 'I' of 'OMNIA'.

Faculdades Adamantinenses Integradas (FAI)

www.fai.com.br

SCATALON, Lilian Passos; GARCIA, Rogério Eduardo; CORREIA, Ronaldo Celso Messias. Introdução a ontologias e suas aplicações. *Omnia Exatas*, v.3, n.2, p.19-28, 2010.

INTRODUÇÃO A ONTOLOGIAS E SUAS APLICAÇÕES

INTRODUCTION TO ONTOLOGIES AND THEIR APPLICATIONS

Lilian Passos Scatalon

Graduanda em Ciência da Computação – Unesp – Presidente Prudente

Rogério Eduardo Garcia

Ronaldo Celso Messias Correia

Professores Doutor – DMEC – Unesp – Presidente Prudente

RESUMO

Diferentes vocabulários e contextos são obstáculos para a comunicação entre pessoas ou sistemas de software. É preciso um entendimento comum sobre o domínio que se conversa, a fim de se obter uma interpretação correta das informações. Uma ontologia modela formalmente a estrutura de um domínio e explicita o seu entendimento compartilhado sob a forma de conceitos e relações que emergem de sua observação. Constitui-se em uma espécie de arcabouço para o mapeamento ao significado da informação sobre a qual se conversa. O rigor formal com que são definidas, por meio de axiomas, permite que sejam processáveis por máquina, implicando na interoperabilidade de sistemas. Estruturado dessa forma, o conhecimento é facilmente transferido entre pessoas ou sistemas de diferentes contextos. Ontologias apresentam diversas aplicações atualmente. São consideradas a infraestrutura da Web Semântica, a qual é composta por recursos Web com significado embutido. Dessa forma, permite que sejam executadas tarefas complexas automaticamente, tirando proveito da comunicação efetiva entre agentes de software da Web. Entre outras aplicações, também vem sendo utilizadas para estruturar o conhecimento gerado por diversas áreas, como Biologia e Engenharia de Software.

Palavras-chave: Ontologia, Representação de Conhecimento.

ABSTRACT

Different vocabularies and contexts are barriers to the communication between people or software systems. It is necessary a common understanding in the domain that is talked about, so it can be obtained a correct interpretation of the information. An ontology formally models the structure of a domain and turn explicit the shared understanding in the form of concepts and relations that emerge from its observation. Constitutes a sort of framework used in the mapping to the meaning of the information that is talked about. The formal accuracy in which they are defined, by means of axioms, allow machine processing, implicating in systems interoperability. Structured this way, the knowledge is easily transferred between people or systems from different contexts. Ontologies present several applications nowadays. They are considered the infra-structure to the Semantic Web, which is composed by Web resources with embedded meaning. Thereby, the automatic execution of complex tasks is allowed, with the benefit from the effective communication between Web software agents. Among other applications, they also have been used to structure the knowledge generated from several areas, like Biology and Software Engineering.

Keywords: Ontology, Knowledge Representation.

INTRODUÇÃO

Pessoas, organizações e sistemas de software precisam se comunicar. Do ponto de vista da Semiótica, o ato de comunicar é a materialização do pensamento (ou sentimento) em signos conhecidos pelas partes envolvidas (emissores e receptores de mensagens). Entretanto, cada um entre as partes envolvidas pode utilizar o seu vocabulário próprio para expressar sua ideia ou então perceber uma mesma realidade de diversos modos, devido a diferentes contextos (pontos de vista da realidade). A consequente falta de **entendimento compartilhado** gera problemas à comunicação entre as partes. Sob uma perspectiva mais técnica da área da

Ciência da Computação, esse problema limita a interoperabilidade de sistemas, uma vez que a troca de mensagem depende da correta expressão e interpretação desta, assim como o potencial para reuso (Uschold e Gruninger, 1996). Ontologias representam o entendimento compartilhado de algum domínio de interesse, mapeando o vocabulário para comunicação e seu correto entendimento, sendo usadas para resolver problemas de comunicação (Uschold e Gruninger, 1996).

Uma ontologia consiste em um artefato no qual se embute uma representação de conhecimento incluindo definições de conceitos de um domínio e relações entre eles, legíveis por computadores. Os níveis de formalidade dessa especificação variam. Entretanto, uma ontologia (formal) tem como componentes básicos classes, suas propriedades e suas instâncias. Noy e McGuinness (2001) apontam as seguintes motivações para se desenvolver ontologias: compartilhar entendimento comum da estrutura da informação, tanto entre pessoas quanto entre agentes de software; permitir reuso de conhecimento do domínio; tornar explícitas suposições sobre o domínio.

O fato de que essa modelagem sobre um domínio seja expressa de maneira formal, faz com que seja possível troca de informações com significado sobre ele, facilitando a interpretação adequada sobre o que se está conversando. Ontologias são utilizadas como um arcabouço na comunicação de agentes de software, possibilitando sua interoperabilidade. Também são consideradas a infra-estrutura da Web Semântica, em que recursos da Web são expressos com significado, permitindo análise automatizada de seu conteúdo.

Do ponto de vista da comunicação entre pessoas, ontologias são utilizadas principalmente para padronização de conceitos. Uschold e Gruninger (1996) apontam que acontece de pesquisadores de campos de pesquisa diferentes, mas relacionados, não conseguirem usar facilmente os resultados uns dos outros. Isso acontece porque os pesquisadores têm uma perspectiva diferente e usam termos diferentes para descrever as mesmas ideias subjacentes. Com o uso de ontologias, o que se conhece sobre um domínio fica explicitamente definido sob a forma de conceitos e relações, com restrições sobre as possibilidades de interpretação a fim de se chegar ao significado pretendido dos conceitos do domínio, eliminando potenciais ambiguidades e interpretações incorretas sobre qualquer conceito, e como consequência, sobre a informação que se transmite baseada nesta definição.

Dada sua importância para a interoperabilidade entre sistemas computacionais, neste artigo são apresentados conceitos sobre ontologias, princípios e utilizações dessas. Para isso, na Seção 2 é apresentado um conceito geral de ontologia. Na Seção 3 são mostrados os seus componentes principais, utilizados para expressá-las. Na Seção 4 é apresentado o processo de desenvolvimento de ontologias. São expostos tipos de aplicações mais comuns para ontologia na Seção 5. Na Seção 6 são apresentadas algumas considerações sobre o uso ontologias.

CONCEITO

O termo **ontologia** pode assumir diversos significados. Um esclarecimento terminológico é dado em Guarino e Giaretta (1995). A principal distinção que deve ser feita é entre **Ontologia**, a disciplina filosófica que lida com a natureza e estrutura da realidade, e **ontologia**, que ainda considerada no sentido filosófico, se refere a um sistema de categorias que descrevem uma certa visão do mundo (Pretorius, 2004). O conceito de ontologia em Ciência da Computação foi adotado da Filosofia por pesquisadores de Inteligência Artificial, e desde então tem sido usado com um novo significado, não completamente desconexo do pertencente ao contexto filosófico (Pretorius, 2004). Em Ciência da Computação, uma ontologia é um tipo especial de objeto de informação ou artefato computacional (Guarino et al., 2009), que incorpora algum tipo de visão do mundo em relação a um dado domínio (Pretorius, 2004).

Ontologias são meios de modelar formalmente a estrutura de um sistema, ou seja, as entidades e relações que emergem de sua observação (Guarino et al., 2009). Studel et al. (1998) declara que uma ontologia é uma especificação **formal** e **explícita** de uma **conceituação compartilhada**. Uma **conceituação** é uma visão abstrata e simplificada do mundo que se deseja representar para algum propósito (Guarino et al., 2009). Tal visão é constituída de conceitos deste mundo e relações entre eles.

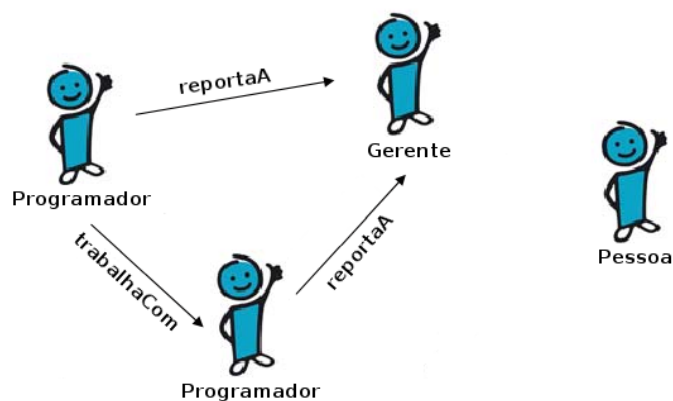


Figura 1 – Recursos humanos em uma empresa de software, adaptado de Guarino et al. (2009)

Para facilitar o entendimento, suponha que tal mundo que se deseja representar (extremamente simplificado) seja a parte de recursos humanos de uma empresa de software (vide Figura 1). Com a observação desse mundo, emergem conceitos e relações importantes entre eles, de acordo com os propósitos de interesse. No exemplo, os conceitos são pessoas, gerentes e programadores. Assim, a conceituação dos recursos humanos desse universo (empresa) em UML é dada pela Figura 2.

Até então, ao elaborar a conceituação, obtém-se uma captura dos conceitos do domínio que compõem a ontologia. A ideia do que vem a ser o domínio de interesse fica expressa de maneira **explícita** sob a forma de conceitos e relações entre eles. Entretanto, deve haver uma **especificação formal** dos conceitos, em um formato que seja legível por máquinas. Para isso, uma ontologia é formalizada por meio de alguma linguagem, com sintaxe e semântica definidas, que provenha a restrição da interpretação de seus conceitos, utilizando-se de **axiomas**. Uma ontologia formal pode ser considerada como um conjunto de tais axiomas.

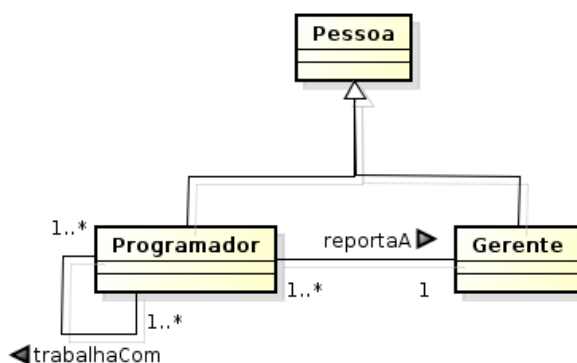


Figura 2 – Conceituação dos recursos humanos de uma empresa de software

O axioma do exemplo, em Lógica de Primeira Ordem, que define o vínculo entre dois programadores que trabalham juntos é dado por:

$$(\forall p1,p2) \text{trabalhaCom}(p1,p2) \rightarrow \text{reportaA}(p1,g) \wedge \text{reportaA}(p2,g)$$

em que $p1$ e $p2$ representam programadores e g o gerente ao qual $p1$ e $p2$ estão subordinados.

Uma especificação formal de uma conceituação não é, necessariamente, uma especificação de uma **conceituação compartilhada** (Guarino et al., 2009). A fim de que o conhecimento seja compartilhado entre as partes que utilizam a ontologia, deve haver concordância sobre o que está sendo comunicado. O conhecimento deve ser consensual entre as partes. Desse aspecto emerge a questão de **compromisso ontológico** (*ontological commitment*).

Considerando dois agentes de software em comunicação, quando um deles se depara com um termo como 'gerente', precisa utilizar os seus modelos internos para vincular esta referência léxica a um conceito do

mundo que ela representa (Pretorius, 2004). Se há um compromisso a uma ontologia, neste caso de recursos humanos de uma empresa de software, então há uma concordância de que 'gerente' se refere a gerentes de projeto de software, e não de um gerente de banco, por exemplo. As atividades de cada um são diferentes, e portanto os conceitos associados a cada um na modelagem dos respectivos domínios também são. Para facilitar essa comunicação entre agentes, sejam de software ou não, a ontologia serve como um conhecimento do contexto que exclui referências incorretas (Pretorius, 2004).

Por fim, após analisar cada um dos conceitos que a definição de Studel et al. (1998) embute, pode-se afirmar que uma ontologia é uma especificação de conhecimento consensual sobre um modelo abstrato de domínio, definida explicitamente em termos de conceitos, suas propriedades e relações por meio de axiomas, possibilitando, assim, que seja legível por máquinas.

COMPONENTES DE UMA ONTOLOGIA

Uma ontologia pode tomar diferentes formas, mas necessariamente inclui um vocabulário de termos e alguma especificação de seus significados (Uschold e Gruninger, 1996). O nível de formalidade de uma ontologia pode variar, mas para os propósitos deste artigo, consideram-se ontologias formais, cujos termos sejam definidos com semântica formal. Os componentes básicos de uma ontologia¹ são **classes**, **propriedades** e **instâncias**. Todas as definições destes componentes foram retiradas de Horridge (2009).

Instâncias representam os objetos do domínio que está sendo representado. Na Figura 3 são apresentadas instâncias do domínio de recursos humanos da empresa de software.

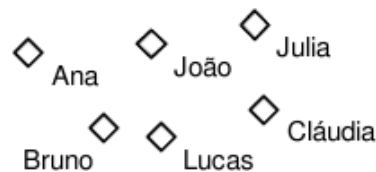


Figura 3 – Representação de instâncias

Propriedades são relações binárias entre instâncias. A propriedade *reportaA* liga a instância Julia à instância João na Figura 4, o que expressa o fato de que Julia é subordinada a João.



Figura 4 – Representação de propriedades

Classes são conjuntos que contêm instâncias. Podem ser consideradas como representações concretas de conceitos do domínio. Por exemplo, a classe Programador contém todos os programadores que trabalham na empresa de software do exemplo.

As classes devem ser organizadas em uma hierarquia. Considerando as classes do exemplo, Programador e Gerente devem ser subclasses de Pessoa, ou seja, as instâncias do domínio que forem programadores ou gerentes, também devem ser pessoas. Subclasses especializam (são subsumidas por) suas superclasses. Na Figura 5 pode-se ver a representação de classes contendo instâncias.

¹ Considerando OWL (*Web Ontology Language*), linguagem padrão W3C. OWL se vale da Lógica Descritiva, um fragmento decidível da Lógica de Primeira Ordem, para expressar seus axiomas.

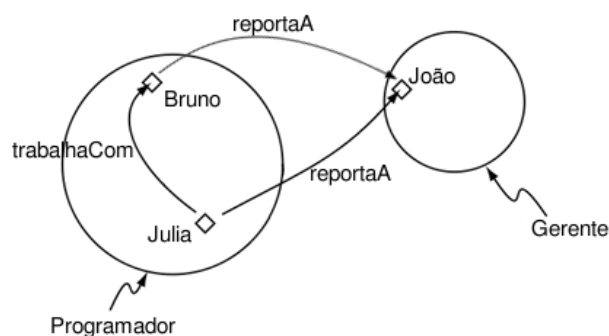


Figura 5 – Representação de classes

Uma classe é definida por meio de descrições formais (matemáticas) que declaram os requisitos de pertinência da classe (para uma instância). Tais requisitos delimitam a interpretação do conceito valendo-se de **restrições de propriedades**. Todos os tipos de restrições descrevem um conjunto que pode conter instâncias, o qual pode ser visto como uma **classe anônima**. Qualquer instância que for membro dessa classe anônima satisfaz a restrição que a descreve.

Exemplos de restrições de propriedade são dados na Tabela 1, tanto em Lógica Descritiva quanto na forma com que devem ser compostas no editor de ontologias Protégé², com a finalidade de mostrar a sintaxe com que são representadas. A restrição de propriedade:

= reportaA 1 Gerente

representa a classe anônima que contém todas as instâncias, sejam programadores ou não, de todo o domínio que sejam subordinadas a exatamente um gerente, ou seja, que apresentem exatamente uma propriedade reportaA ligando-as a uma instância de Gerente. Ao descrever uma classe usando restrições declaram-se as classes anônimas que são suas superclasses. Existem três tipos de restrições:

- restrições de quantificação;
- restrições de cardinalidade;
- restrições hasValue.

Tabela 1 – Exemplos de restrições de propriedade

Tipo de Restrição		Lógica Descritiva	Protégé
Restrição de Quantificação	Existencial	\exists prop ClasseA	prop some ClasseA
	Universal	\forall prop ClasseB	prop only ClasseB
Restrição de Cardinalidade	Mínima	\geq prop 3	prop min 3
	Máxima	\leq prop 4	prop max 4
	-	= prop 2	prop exactly 2
	Qualificada	= prop 2 ClasseC	prop exactly 2 ClasseC
Restrição hasValue	-	prop \ni InstanciaX	prop value InstanciaX

Para definir a classe Programador, deve-se limitar as suas possibilidades de interpretação no domínio, traduzir o significado pretendido para ela em termos de suas definições formais. Devem ser especificadas as suas restrições de propriedades, como pode ser visto abaixo:

= reportaA 1 Gerente

\exists trabalhaCom Programador

A primeira restrição corresponde a uma restrição de cardinalidade qualificada, que especifica a quantidade exata de relações que podem existir a instâncias da classe especificada, neste caso Gerente. A segunda é uma restrição existencial, a qual determina que deve existir pelo menos uma relação por meio da propriedade com

² Protégé é um editor de ontologias que pode ser obtido a partir de <http://protege.stanford.edu/>

instâncias da classe especificada (Programador). Essa restrição representa uma classe anônima de todas as instâncias do domínio que trabalham com pelo menos um programador. A descrição da classe Programador é dada pela interseção entre suas superclasses, Pessoa e as duas superclasses anônimas (as duas restrições). Na Figura 6 pode-se observar esta definição de classe no Protégé.

As restrições mostradas até aqui agem sobre propriedades que ligam instâncias, ou **propriedades de objeto**. De maneira similar, podem ser elaboradas restrições para **propriedades de tipo de dado**, que relacionam uma instância a um tipo de dado (inteiro, string, etc). Uma restrição deste tipo que a classe Pessoa pode apresentar é “= temNome string”.

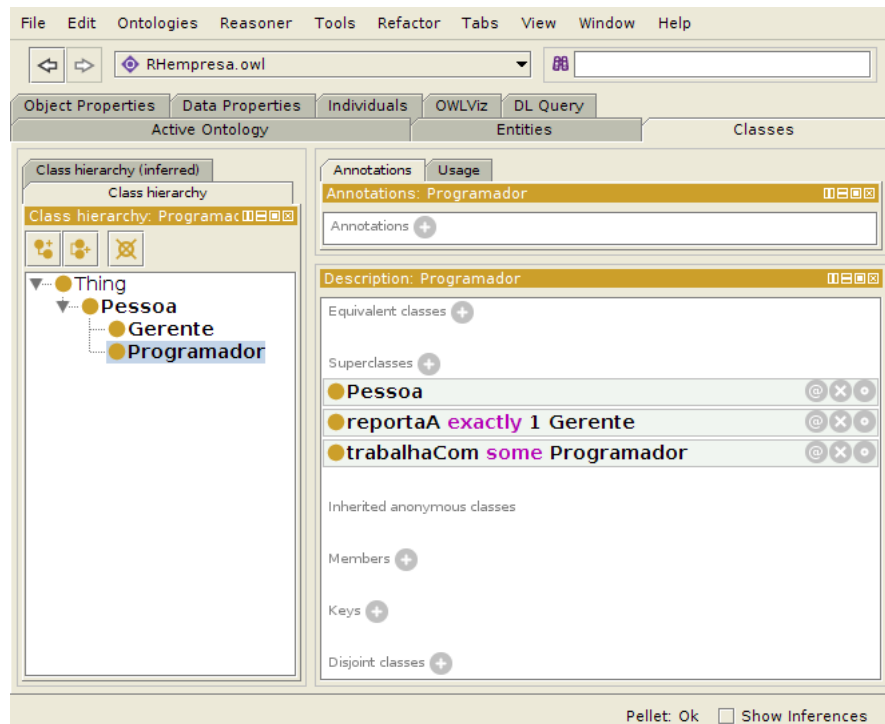


Figura 6 – Descrição da classe Programador no editor Protégé

DESENVOLVIMENTO DE ONTOLOGIAS

Segundo Noy e McGuinness (2001), desenvolver uma ontologia inclui:

- definir classes na ontologia,
- arranjar as classes em uma hierarquia (subclasse-superclasse),
- definir propriedades e descrever valores permitidos a elas,
- preencher os valores de propriedades para instâncias.

Entretanto, elaborar ontologias envolve mais do que apenas definir os conceitos da ontologia de maneira formal. Deve-se determinar o escopo do domínio, analisá-lo para capturar a sua conceituação, considerar o reuso de ontologias já existentes, entre outras atividades que trouxeram o surgimento da Engenharia de Ontologias, que lida com o seu desenvolvimento prático. Há diversas propostas de processos de desenvolvimento de ontologias, como em Uschold e Gruninger (1996), Noy e McGuinness (2001) e Rautenberg et al. (2009). O último se compõe das melhores práticas de outras metodologias e está dividido em cinco grandes atividades. São elas:

1. **Especificação:** avaliar os custos do desenvolvimento da ontologia.
2. **Conceituação:** descrever um modelo conceitual do domínio de discurso.
3. **Formalização:** transformar o modelo conceitual em um modelo formal, passível de ser implementado.
4. **Implementação:** implementar a ontologia formalizada em uma linguagem de representação adequada.

5. **Avaliação:** a ontologia é validada quanto ao entendimento aceito sobre o domínio em fontes de conhecimento. Verifica-se a coerência do conhecimento representado na ontologia e certifica-se de sua utilidade.

APLICAÇÕES

O aspecto de compartilhamento é fundamental para a noção de ontologias, como se percebe em Gruber (1993). Ontologias provêm um arcabouço explícito para discutir sobre alguma realidade (Noy, 2005). Dessa forma, ontologias podem ser utilizadas como uma infra-estrutura para aplicações, que por sua vez se beneficiam da concordância quanto aos conceitos representados que um vocabulário compartilhado oferece.

A aplicação mais notável está na Web Semântica: uma organização de conteúdo da Web que pode ser processado por máquinas (Noy, 2005). Nessa abordagem, um recurso de informação deve ser representado de maneira a incorporar o seu significado, apresentando metadados que permitam que agentes de software processem os dados. Ontologias são consideradas a infra-estrutura da Web Semântica. Berners-Lee et al. (2001) descrevem uma visão de como seria para marcar uma consulta médica *on-line* com o estabelecimento da Web Semântica: os agentes de software encontrariam médicos vinculados ao plano de saúde do usuário e marcariam a consulta de forma a satisfazer as restrições de horário tanto do médico quanto do usuário, todas essas atividades realizadas automaticamente.

OWL, a linguagem padrão atual para desenvolvimento de ontologias, estende padrões já existentes da Web, como XML, RDF e RDFS. Um guia da linguagem é dado em Smith et al. (2004) e a semântica formal está definida em Patel-Schneider et al. (2004). Há grande variedade de ferramentas disponíveis para OWL, uma das possíveis razões para sua ampla adoção em meios acadêmicos e de mercado (Horridge e Bechhofer, 2009). Essas ferramentas têm suportado a criação e edição de ontologias OWL (Protégé³, Swoop⁴), a computação de suas vinculações ou *reasoning*⁵ (Hermit⁶, Pellet⁷) e o uso de ontologias em aplicações (Jena Ontology API⁸ e OWLAPI⁹).

Ontologias também vêm sendo utilizadas em Engenharia de Software com a finalidade de estabelecer um entendimento comum sobre os conceitos da área, para reuso e compartilhamento de conhecimento (Kitchenham et al. (1999); Falbo et al. (1999); Falbo et al. (2005); Barbosa et al. (2006); Biolchini et al. (2007); Garcia et al. (2008)).

Por fim, entre diversas outras aplicações, o uso de ontologias tem se tornado fundamental em Bioinformática, com o objetivo de padronização de conceitos de Biologia. Embora o que se observa com mais frequência seja a construção de vocabulários controlados (estruturas menos formais), o objetivo de ontologias neste domínio tem caminhado em direção a favorecer suporte computacional e análise de dados – por exemplo, dados obtidos em experimentos (Stevens e Lord, 2009).

CONSIDERAÇÕES FINAIS

O uso de ontologias é motivado por contornar problemas de comunicação, por fornecerem um entendimento compartilhado sobre o domínio de que tratam as informações trocadas entre as partes. Diversas aplicações tiram proveito desse aspecto para padronização de conceitos, com a finalidade de facilitar a comunicação, seja entre pessoas ou entre sistemas de software, acerca de determinado domínio. O rigor formal com que são especificadas permite o processamento de ontologias por máquinas, permitindo que sejam trocadas

³ <http://protege.stanford.edu>

⁴ <http://code.google.com/p/swoop>

⁵ Um reasoner é um sistema que computa vinculações (*entailments*) inferidas do que é declarado na ontologia.

⁶ <http://hermit-reasoner.com>

⁷ <http://clarkparsia.com/pellet>

⁸ <http://jena.sourceforge.net/ontology>

⁹ <http://owlapi.sourceforge.net>

informações com significado embutido, ocasionando a interpretação de tais informações por agentes de software da maneira pretendida.

A principal aplicação de ontologias é a representação de recursos da Web Semântica. Entretanto, para possibilitar essa comunicação entre agentes de domínios diferentes, que, portanto, utilizam ontologias diferentes, é necessário primeiro encontrar as **correspondências** entre conceitos destas ontologias. Esse processo é chamado de **mapeamento de ontologias**, e apresenta muitas dificuldades tanto em determinar tais correspondências entre classes e propriedades quanto encontrá-las automaticamente (Noy, 2009). Ontologias também se tornaram fundamentais para a construção de sistemas inteligentes de suporte à decisão, de simulação, de integração de dados, de recuperação de informação e de linguagem natural¹⁰ (Noy, 2005).

Estão disponíveis diversos repositórios de ontologias atualmente. Alguns de domínio específico como o *Open Biological and Biomedical Ontologies*¹¹, ou de propósito mais geral como o *TONES Ontology Repository*¹².

REFERÊNCIAS

BARBOSA, E. F.; NAKAGAWA, E. Y.; MALDONADO, J. C. Towards the establishment of an ontology of software testing. In: SEKE (Software Engineering & Knowledge Engineering), 2006, San Francisco, CA, p. 522–525.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. *Scientific American*, v. 284, n. 5, p. 34–43, 2001.

BIOLCHINI, J. et al. Scientific research ontology to support systematic review. *Advanced Engineering Informatics*, v. 21, n. 2, p. 133–151, 2007.

FALBO, R.; RUY, F. B.; MORO, R. D. Using ontologies to improve knowledge integration in software engineering environments. In: *World Multiconference on Systemics, Cybernetics and Informatics*, 1998, Orlando. *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, 1999.

FALBO, R.; RUY, F. B.; MORO, R. D. Using ontologies to add semantics to a software engineering environment. In: SEKE (Software Engineering & Knowledge Engineering), 2005, Taipei, Taiwan, Republic of China, p. 151–156.

GARCIA, R. E. et al. An Ontology for Controlled Experiments on Software Engineering. In: SEKE (Software Engineering & Knowledge Engineering), 2008, San Francisco, CA, USA, p. 685–690.

GRUBER, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*. v. 43, n. 5-6, p. 907–928, 1993.

GUARINO, N.; GIARETTA, P. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: MARS, N. J. I (ed.). *Towards very large knowledge bases: Knowledge building and knowledge sharing*. Amsterdam: IOS Press, 1995, p. 25–32.

GUARINO, N.; OBERLE, D.; STAAB, S. What is an Ontology? In S. STAAB; R. STUDER (eds.), *Handbook on Ontologies*, Second Edition. *International handbooks on information systems*. Berlin: Springer Verlag, 2009, p. 201–221.

HORRIDGE, M. *A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*, 2009. Disponível em: <<http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial>> Acesso em: 4 de outubro de 2010.

¹⁰ Como por exemplo, extração de informação de textos em linguagem natural.

¹¹ <http://www.obofoundry.org>

¹² <http://owl.cs.manchester.ac.uk/repository>

HORRIDGE, M; BECHHOFER, S. The OWL API: A Java API for Working with OWL 2 Ontologies. In OWLED (OWL: Experiences and Directions), 2009. Disponível em: <<http://www.webont.org/owled/2009>>. Acesso em: 4 de outubro de 2010.

KITCHENHAM, B. A. et al. Towards an Ontology of software maintenance. Journal of Software Maintenance, vol. 11, n. 6, p. 365-389, 1999.

NOY, N.; MC GUINNESS, D. L. Ontology development 101: A guide to creating your first ontology. Relatório Técnico, Stanford University, 2001. Disponível em: <http://protege.stanford.edu/publications/ontology_development/ontology101.pdf>. Acesso em: 4 de outubro de 2010.

NOY, N. Order from Chaos. ACM Queue, v. 3, n. 8, p. 42-49, 2005.

NOY, N. Ontology Mapping. In S. STAAB; R. STUDER (eds.), Handbook on Ontologies, Second Edition. International handbooks on information systems. Berlin: Springer Verlag, 2009, p. 573-590.

PATEL-SCHNEIDER, P.; HAYES, P.; HORROCKS, I. OWL Web Ontology Language Semantics and Abstract Syntax. Recomendação W3C, 2004. Disponível em <<http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>>. Acesso em: 4 de outubro de 2010.

PRETORIUS, A. J. Ontologies - Introduction and Overview, 2004. Disponível em <http://www.starlab.vub.ac.be/teaching/Ontologies_Intr_Overv.pdf>. Acesso em: 4 de outubro de 2010.

SMITH, M. K.; WELTY, C.; L., MC GUINNESS, D. L. OWL Web Ontology Language Guide. Recomendação W3C, 2004. Disponível em <<http://www.w3.org/TR/owl-guide>>. Acesso em: 4 de outubro de 2010.

STEVENS, R.; LORD, P. Application of Ontologies in Bioinformatics. In: S. STAAB and R. STUDER (eds.), Handbook on Ontologies, Second Edition. International handbooks on information systems. Berlin: Springer Verlag, 2009, p. 735-756.

STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering, v. 25, n. 1-2, p. 161-197, 1998.

USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, methods and applications. Knowledge Engineering Review, v. 11, p. 93-136, 1996.