

Ontologias

MARIANNA ARAÚJO

Roteiro

Motivação

Conceito

Tipos

Linguagens de Ontologia

SPARQL

Apresentação de Ferramentas

Modelagem de uma Ontologia com Protégé

Referencias

Motivação

Aumento exponencial dos dados na *Web*

Ausência de definição semântica das informações

Usuário responsável por classificar quais informações são úteis para um domínio de conhecimento

Conceito

Origem na Filosofia:

- Teoria sobre a existência da natureza, sobre que tipos de coisas existem ou o que dizer sobre o mundo.
- Parte da filosofia que trata do ser enquanto ser, isto é, do ser concebido como tendo uma natureza comum que é inerente a todos a cada um dos seres.

Conceito

Na computação:

- "Uma especificação explícita de uma conceitualização." (Gruber, 1993)
- "Uma especificação formal de uma conceitualização compartilhada." (Borst, 1997)
- "Uma especificação formal e explícita de uma conceitualização compartilhada." (Studer et al, 1998)

Ontologias

Para que servem?

- Melhorar a recuperação da informação ao organizar o conteúdo de fontes de dados que compõem um domínio;
- Capacitar programas a recuperar somente as paginas relevantes para o usuário;

O uso de **Ontologias** visa à organização de informações em um domínio do conhecimento e o melhoramento da exatidão de buscas da *Web*

Ontologias

A relevância das ontologias para *Web* deve ser tratada sob **três** aspectos:

- Identificação de contexto
- Fornecimento de definições compartilhadas
- Reuso de Ontologias

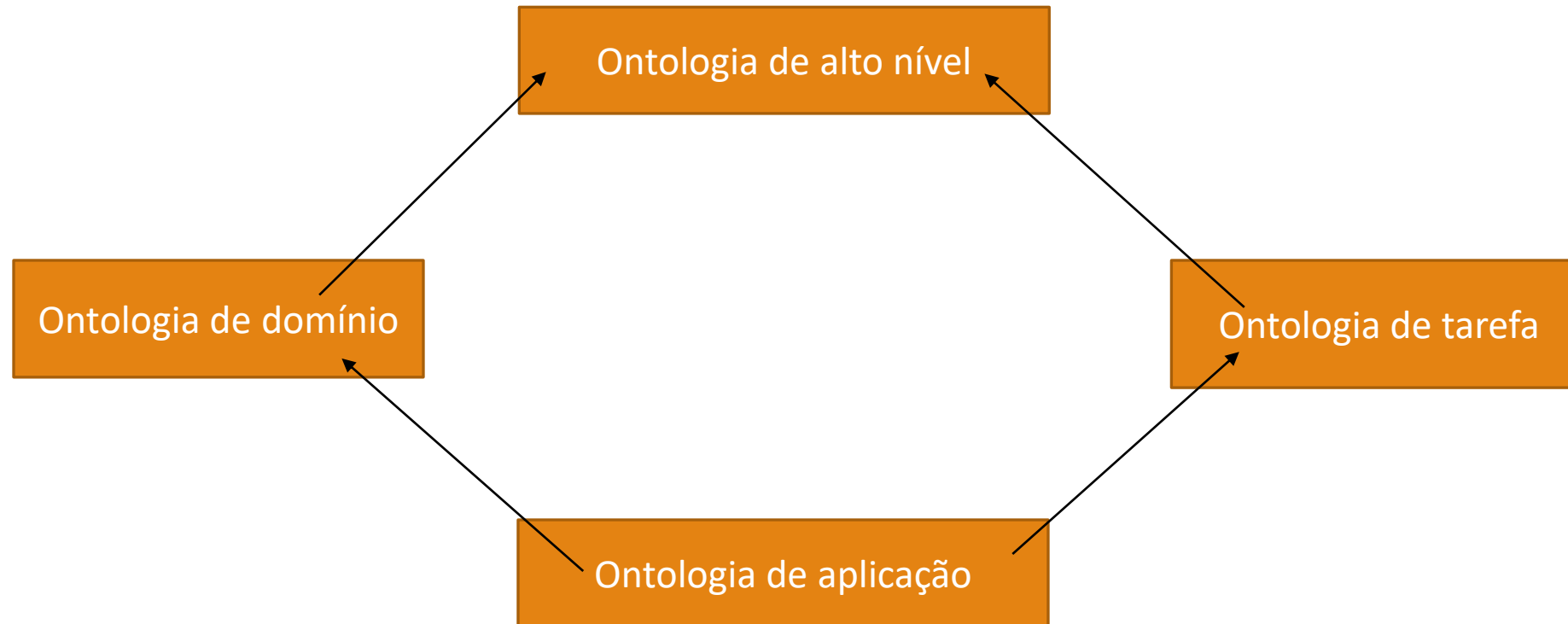
Se as ontologias forem construídas levando em consideração estes aspectos, elas podem ser usadas em buscas baseadas em semântica, facilitando a comunicação entre diferentes tipos de agentes.

Tipos de ontologias

Existem tipos diferentes de ontologias que suportam vários usos delas:

- **Ontologia de alto nível:** entidades genéricas localizadas em um nível de abstração mais alto em relação as entidades de domínio.
- **Ontologia de domínio e tarefa** – representa respectivamente, conhecimento sobre um domínio específico, ou conhecimento sobre uma tarefa específica.
- **Ontologia de aplicação:** é aquela que descreve conceitos dependendo de um domínio e tarefa específicos.

Tipos de ontologias



Linguagens de Ontologias

O uso de ontologias no contexto da Web Semântica requer uma linguagem compatível com a Web.

Dentre as diferentes linguagens de ontologias existentes, pode-se citar:

- SHOE – *Simple HTML Ontology Extensions*
- XOL – *Ontology Exchange Language*
- OIL – *Ontology Inference Layer*
- DAML – *DARPA Agent Markup Language*
- DAML + OIL – Combinação das duas ultimas
- **OWL – *Web Ontology Language***

OWL – Web Ontology Language

Linguagem para definição e instanciação de ontologias Web.

Possui três sub-linguagens incrementais:

- **OWL *Lite***: sub-linguagem OWL DL que usa somente algumas características da linguagem OWL e possui mais limitações do que OWL DL ou OWL *Full*.
- **OWL DL**: inclui todas as construções da linguagem OWL e é usada por usuários que queiram o máximo de expressividade. A sigla DL possui correspondência com a lógica descritiva.
- **OWL *Full***: usada por usuários que queiram o máximo de expressividade e independência sintática de RDF, sem garantia computacional. OWL *Full* e a OWL DL suportam o mesmo conjunto de construções da linguagem OWL.

OWL – Web Ontology Language

Cada uma destas sub-linguagens apresentadas é uma extensão de sua predecessora.

- OWL *Lite* é uma ontologia válida em OWL DL
- OWL DL é uma ontologia válida em OWL *Full*

Esta relação não é simétrica

Todo documento OWL é um documento RDF e todo documento RDF é um documento OWL *Full*

- Nem todo documento RDF é documento OWL *Lite* ou OWL DL válido.

Estrutura das Ontologias

OWL descreve, de um modo natural, classes e relacionamentos entre elas em documentos e aplicações Web.

Os termos usados em uma ontologia, são escritos de forma a evitar a ambiguidade e usados por agentes de software.

Estrutura das Ontologias

Indicação de qual vocabulário esta sendo usado

- Componente inicial de uma ontologia incluindo um conjunto de *namespaces* XML contidos na *tag* `rdf:RDF`

Código 1 – Declaração de *namespace* em OWL.

```
1 <rdf:RDF
2   xmlns = "http://www.mindswap.org/2003/owl/swint/terrorism#"
3   xmlns:base = "http://www.mindswap.org/2003/owl/swint
4                                     /terrorism#"
5   xmlns:terr = "http://www.mindswap.org/2003/owl/swint
6                                     /terrorism#"
7   xmlns:owl = "http://www.w3.org/2002/07/owl#"
8   xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
9   xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
10  xmlns:xsd = "http://www.w3.org/2001/XMLSchema#"
11 >
```

Estrutura das Ontologias

Seguindo a recomendação de McGuinness, Smith e Welty, é interessante o uso de entidades XML.

Código 2 – Declaração de entidades XML em OWL.

```
1  <!DOCTYPE rdf:RDF [  
2    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">  
3    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">  
4    <!ENTITY owl "http://www.w3.org/2002/07/owl#">  
5    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">  
6    <!ENTITY terr "http://www.mindswap.org/2003/owl/swint  
7                                     /terrorism#">  
8    <!ENTITY pers "http://www.mindswap.org/2003/owl/swint  
9                                     /person#">  
10 ]>
```

Cabeçalho das ontologias

Informações sobre a ontologia

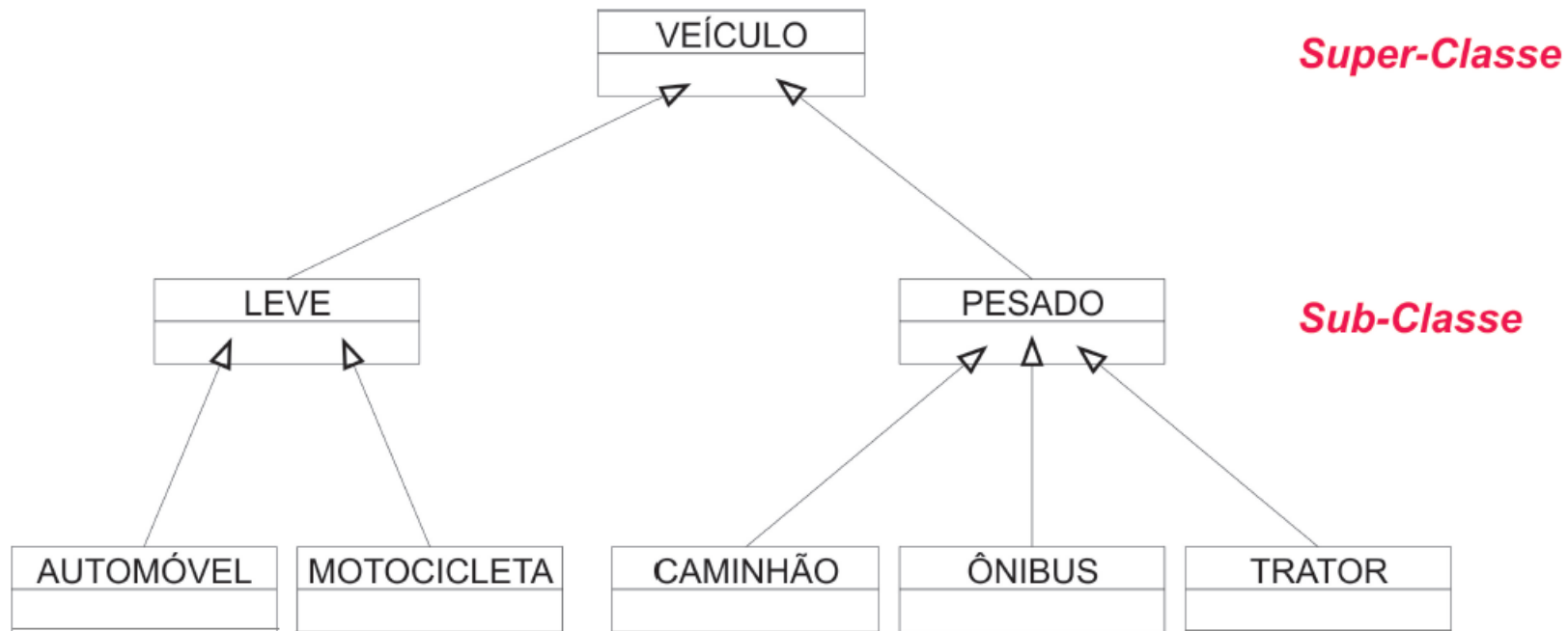
Código 3 – Cabeçalho de uma ontologia.

```
1 <owl:Ontology rdf:about="">
2   <owl:versionInfo> 05/01/2005 20:41:52 </owl:versionInfo>
3   <owl:priorVersion
4     rdf:resource="http://www.mindswap.org/2002/owl/swint
5                                     /terrorism#" />
6   <rdfs:comment>This file specify a OWL ontology</rdfs:comment>
7   <owl:imports rdf:resource="http://www.mindswap.org/2003
8                                     /owl/swint/person#" />
9   <rdfs:label> Terrorism Ontology </rdfs:label>
10   ...
11 </owl:Ontology>
12   ...
13 </rdf:RDF>
```

Construção de Ontologias

Elementos básicos de uma ontologia OWL

- Classes



Construção de Ontologias

Relações

- Representam o tipo de interação entre os elementos das classes.

Axiomas

- Utilizadas para modelar sentenças sempre verdadeiras.

Instâncias

- Os indivíduos que estão associados a uma classe.

Funções

- Eventos que podem ocorrer no contexto da ontologia.

Construção de Ontologias

Cada individuo na OWL é membro da classe *owl:Thing*.

Classe *owl:Nothing* é subclasse de todas as classes OWL.

Definição de uma classe em OWL:

- `<owl:Class rdf:ID="Organization">`

A sintaxe *rdf: ID="Organization"* é usada para nomear a classe.

Construção de Ontologias

Hierarquia de classes podem ser criadas usando-se a construção

- *rdfs:subClassOf*

Código 4 – Hierarquia de classes.

```
1 <owl:Class rdf:ID="TerroristOrganization">
2   <rdfs:subClassOf rdf:resource="#Organization"/>
3   ...
4 </owl:Class>
```

SPARQL – Linguagem de Consulta em Ontologias

Padrão

SPARQL 1.0 se tornou um padrão em janeiro, 2008:

- **SPARQL 1.0 Query Language** – para consultar padrões em dados RDF
- **SPARQL 1.0 Protocol** – para enviar consultas por HTTP
- **SPARQL Results XML Format** – formato dos resultados em XML
- **SPARQL Results JSON Format** – formato dos resultados em JSON

SPARQL – Linguagem de Consulta em Ontologias

Padrão

SPARQL 1.1 se tornou um padrão em Março, 2013:

Versões atualizadas 1.1 da consulta SPARQL e do Protocolo SPARQL

- **SPARQL 1.1 Update** – para inserção, remoção e modificação de dados RDF •
- **SPARQL 1.1 Graph Store HTTP Protocol** – acesso RESTful de grafos RDF
- **SPARQL 1.1 Service Descriptions** – descrição de endpoints SPARQL
- **SPARQL 1.1 Entailments** – como combinar inferência com SPARQL
- **SPARQL 1.1 Basic Federated Query** - consultando vários endpoints de uma vez
- **SPARQL Results CSV/TSV Formats** – formato dos resultados em CSV/TSV

Conceitos

RDF é um modelo de dados de grafos baseado em triplas de sujeito, predicado e objeto:



Sujeitos, predicados e objetos são representados por URIs

Objetos podem também ser literais: strings, inteiros, booleanos, etc.

Sintaxe:

- **URIs:** <http://example.com/resource> ou prefixo:nome
- **Literais:** "string pura" "13.4"^^xsd:float ou "string com linguagem"@pt
- **Tripla:** prefixo:sujeito outro_prefixo:predicado "objeto"

Estrutura de uma Consulta SPARQL

declarações de prefixo

PREFIX foo: <http://example.com/resources/>

...

definição de conjunto de dados

FROM ...

cláusula de resultado

SELECT ...

padrão de consulta

WHERE {

...

}

modificadores de consulta

ORDER BY ...

Exemplos de uma consulta

Ontologia Viagem:

Consultar todos os voos da ontologia:

```
SELECT ?voo  
WHERE{ ?voo rdf:type :Voo }
```

- **rdf:type** significa “é um tipo de”;
- Seleciona todas as instâncias onde essas instâncias são um tipo de Voo (Classe Voo da ontologia).

Exemplos de uma consulta

Consultar todas as passagens:

```
SELECT ?passagem  
WHERE{ ?passagem rdf:type :Passagem}
```

Retorna na variável todas as instâncias pertencentes a classe Passagem.

Ferramentas para Construção de Ontologias

Apollo

Swoop

Protégé

OntoStudio

TopBraid

Protégé

Open-source

Desenvolvido pela Universidade de Stanford

Download gratuito

Permite construir ontologias usando OWL

Permite a restrição de valores de variáveis

Permite o relacionamento entre classes e as propriedades desses relacionamentos.

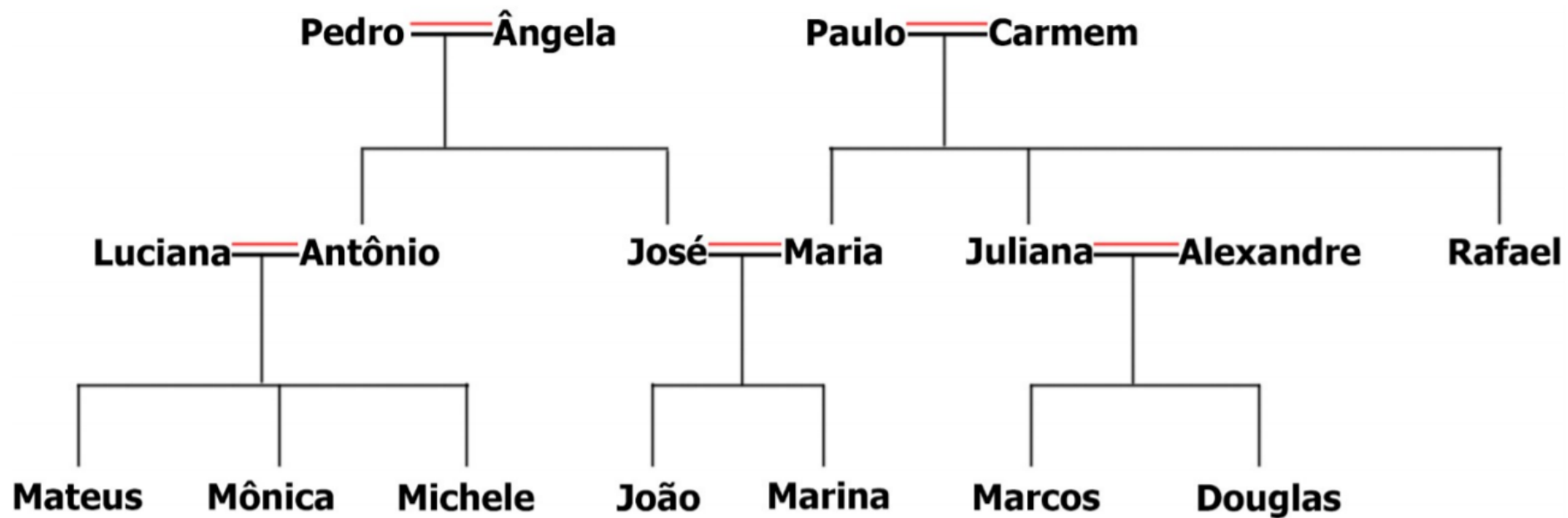
Protégé

Componentes de uma ontologia OWL

- **Individuals:** objetos, porções individuais e únicas do domínio. Representam **instâncias**.
- **Classes:** Conjuntos de indivíduos com propriedades comuns. Representam **conceitos**.
- **Properties:** Propriedades
 - **Object Properties:** Relações binárias entre indivíduos. Representam **relações**.
 - **Data properties:** Relações entre indivíduos e valores de atributos. Representam **atributos**.

Protégé

Modelagem de uma ontologia de relações familiares



Modelagem de uma ontologia de relações familiares

Identificação dos principais conceitos

- Organização em taxonomias de conceitos

Identificação das propriedades

- Relações (verbos)
- Atributos (adjetivos)

Definição de conceitos

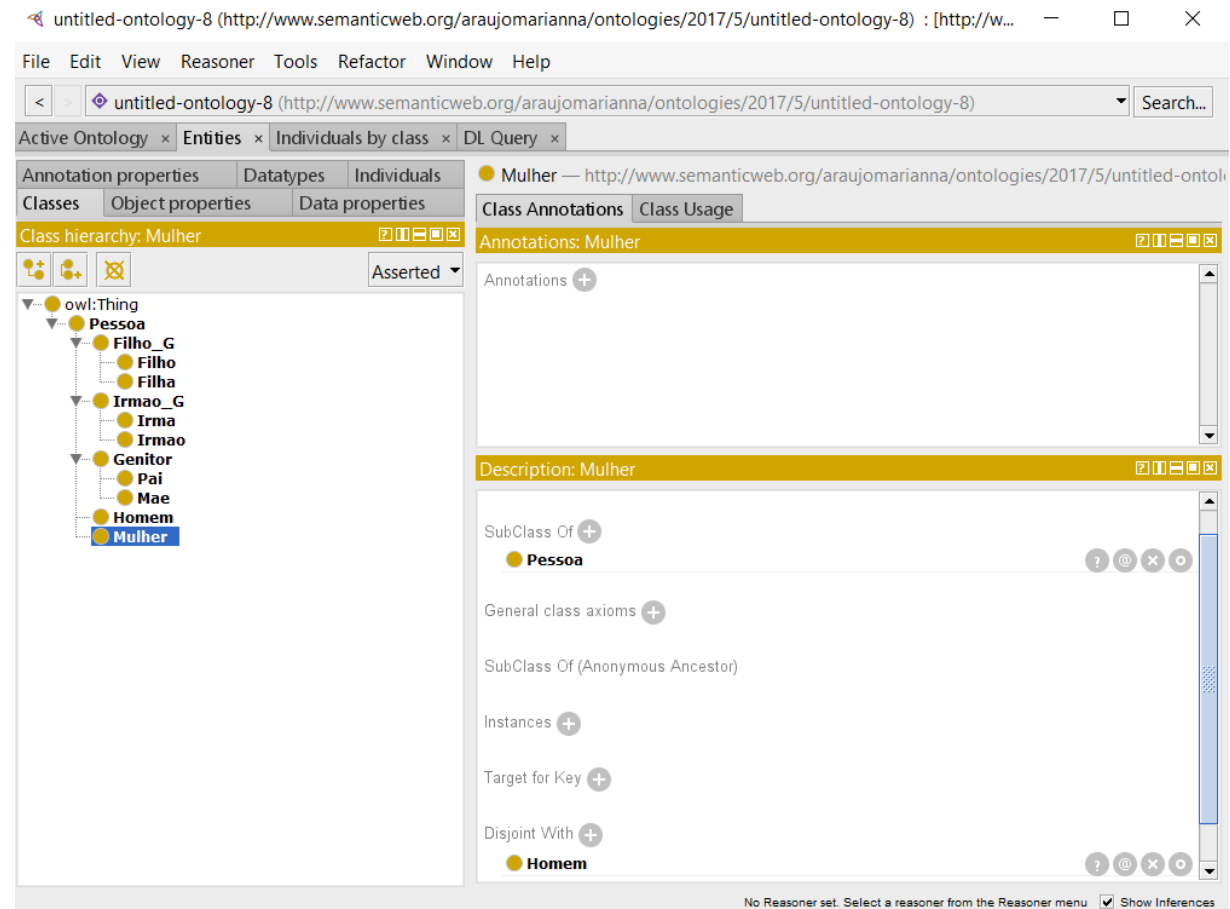
- Condições necessárias e suficientes para um indivíduo pertencer a uma classe

Modelagem de uma ontologia de relações familiares

Principais conceitos do domínio:

- Pessoa
 - Todas as relações familiares são mantidas entre pessoas
- Homem, Mulher
 - Todas as pessoas são homens ou mulheres
 - Quando duas classes representam **conjuntos disjuntos** de indivíduos (não podem estar nas duas simultaneamente), é preciso declarar isto **explicitamente**.

Modelagem de uma ontologia de relações familiares



Modelagem de uma ontologia de relações familiares

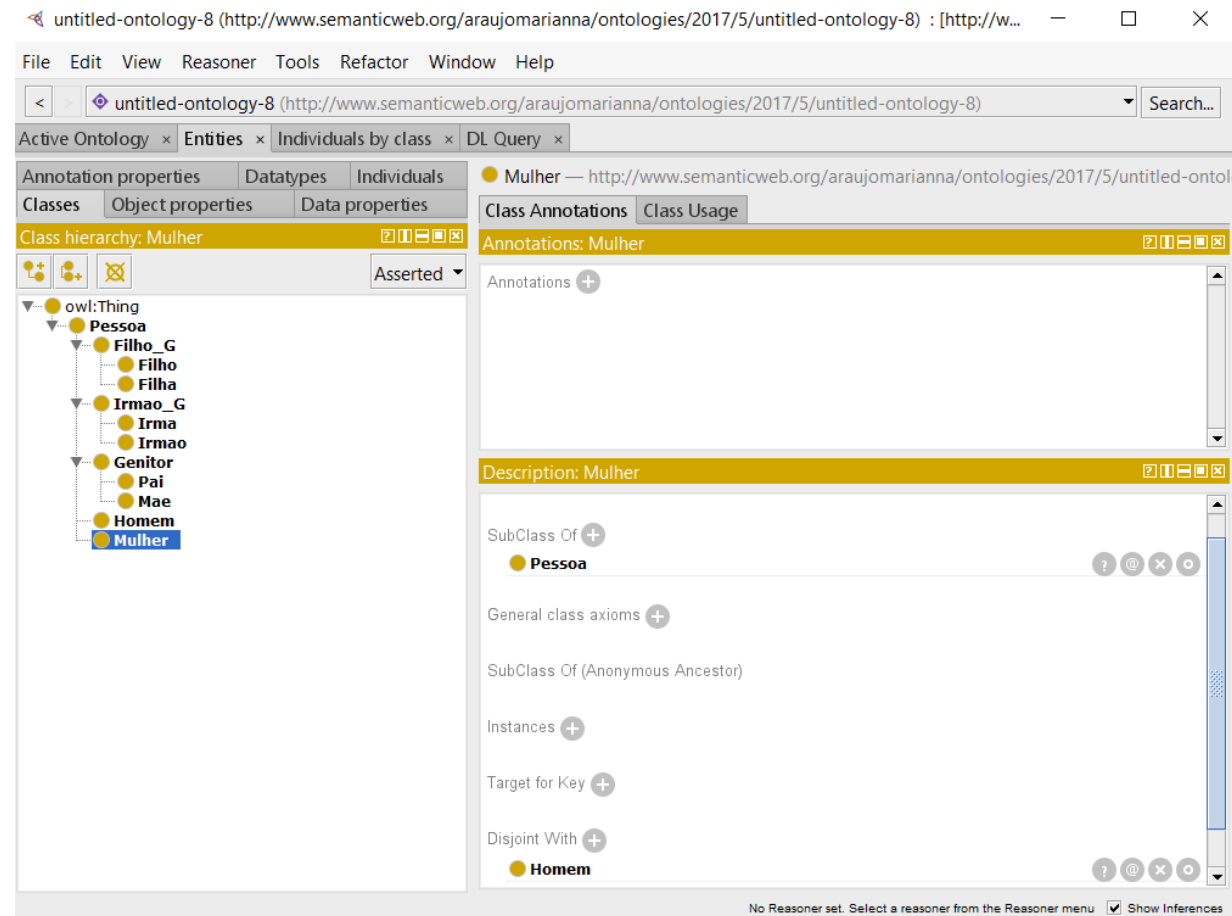
Outros conceitos importantes

- Pai, mãe, filho, filha, irmão, irmã, etc

Alguns pares de conceitos podem ser vistos como disjunções (com base no sexo), de classes mais gerais.

- Genitor (parent), generaliza Pai (father) e Mãe (mother)
- Filho (child), generaliza Filho (son) e Filha (daughter)

Modelagem de uma ontologia de relações familiares



Modelagem de uma ontologia de relações familiares

Atributos (*data type properties*)

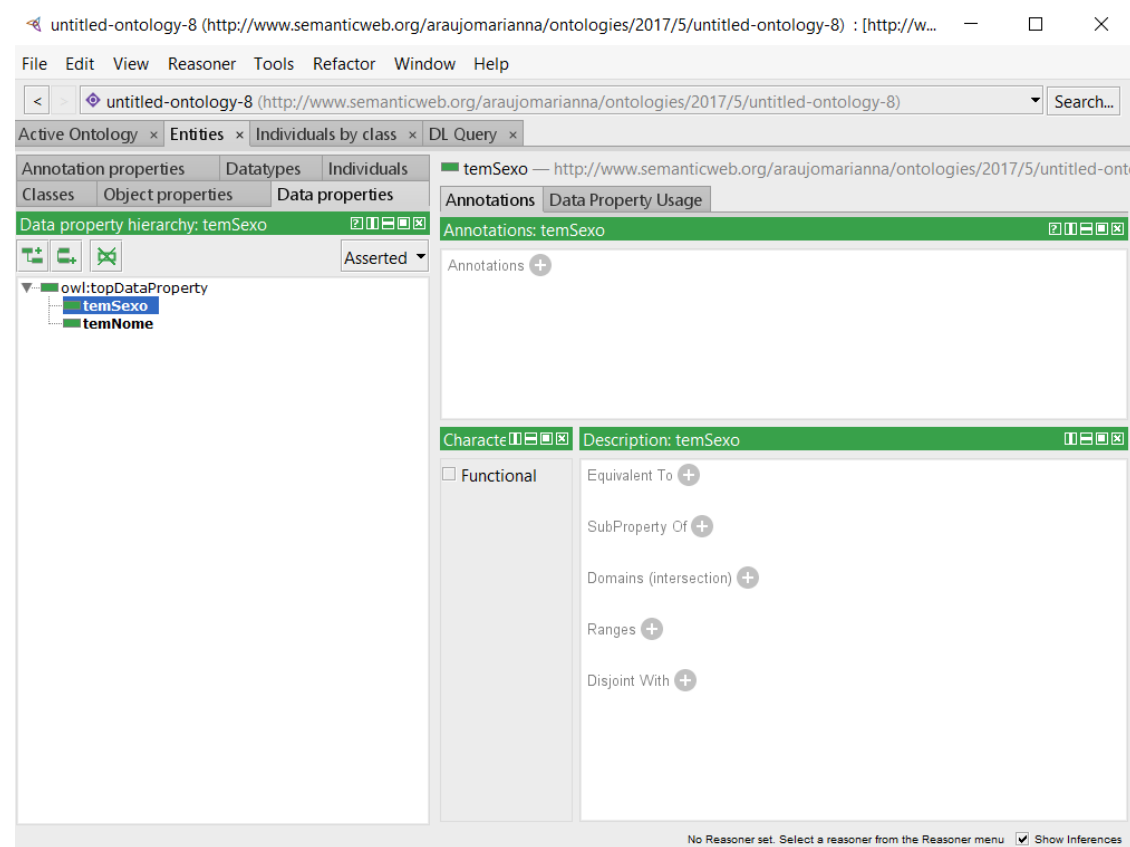
- É importante especificar:
 - Domínio (*domain*):
 - Classes de indivíduos que podem ser descritos por este atributo.
 - Escopo (*range*):
 - Tipos de dados que são admitidos como valores possíveis do atributo

Modelagem de uma ontologia de relações familiares

Atributos (*data type properties*) importantes no domínio

- Neste domínio, todos os conceitos herdam atributos que descrevem Pessoa.
 - Tem sexo
 - Domínio: Pessoa
 - Escopo: String
 - Tem nome
 - Domínio: Pessoa
 - Escopo: String

Modelagem de uma ontologia de relações familiares



Modelagem de uma ontologia de relações familiares

Relações (*object properties*) relacionam indivíduos de certos domínios (*domains*) a indivíduos de certos escopos (*ranges*).

- É importante especificar os domínios e escopos das relações

Modelagem de uma ontologia de relações familiares

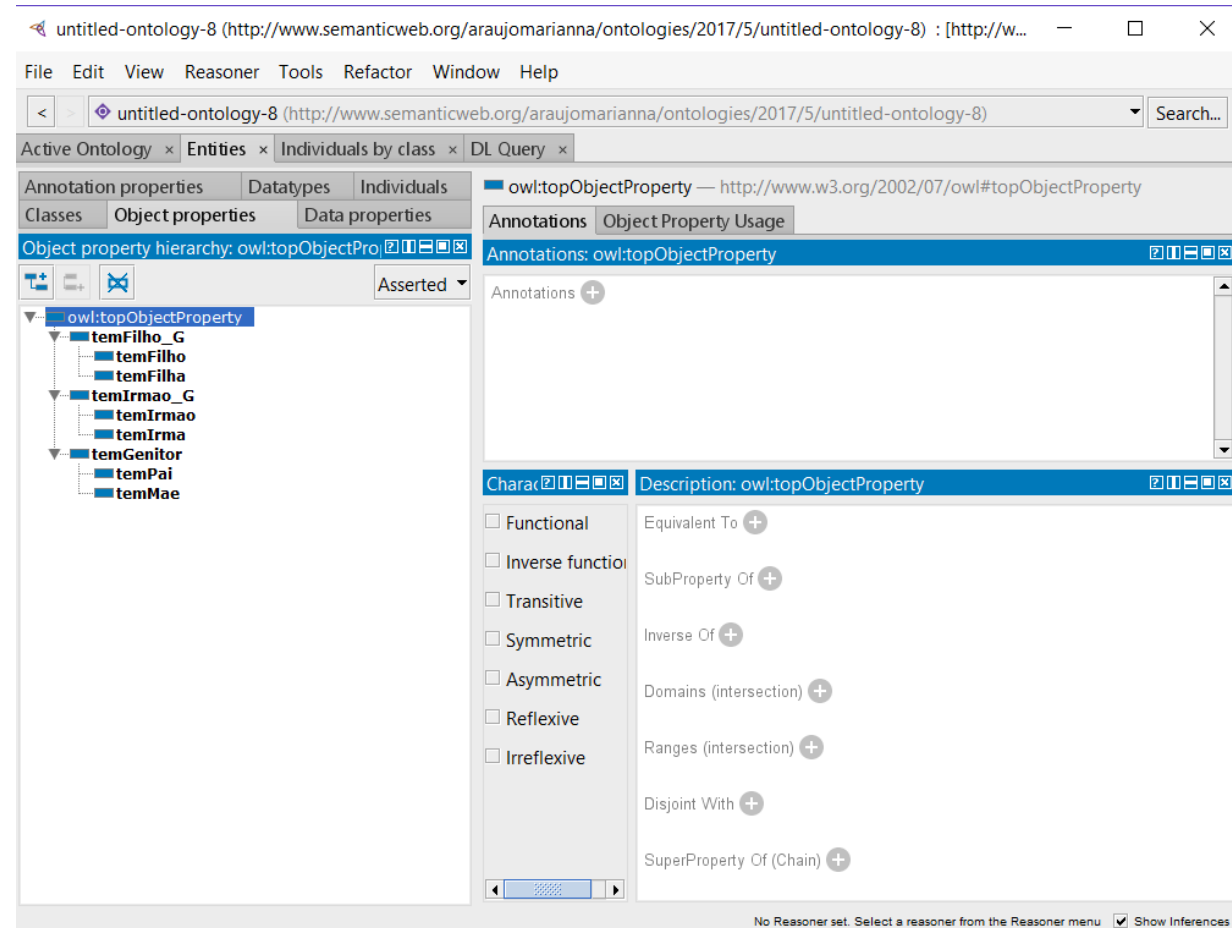
Relações (*object properties*) importantes no domínio:

- Tem filho, tem filha, tem irmão, tem irmã, tem pai, tem mãe, etc

É possível definir hierarquias de relações

- Algumas dessas relações podem ser generalizadas por outras mais gerais
 - temFilho_G (hasChild)
 - temIrmão_G (hasSibling)
 - temGenitor (hasParent)

Modelagem de uma ontologia de relações familiares



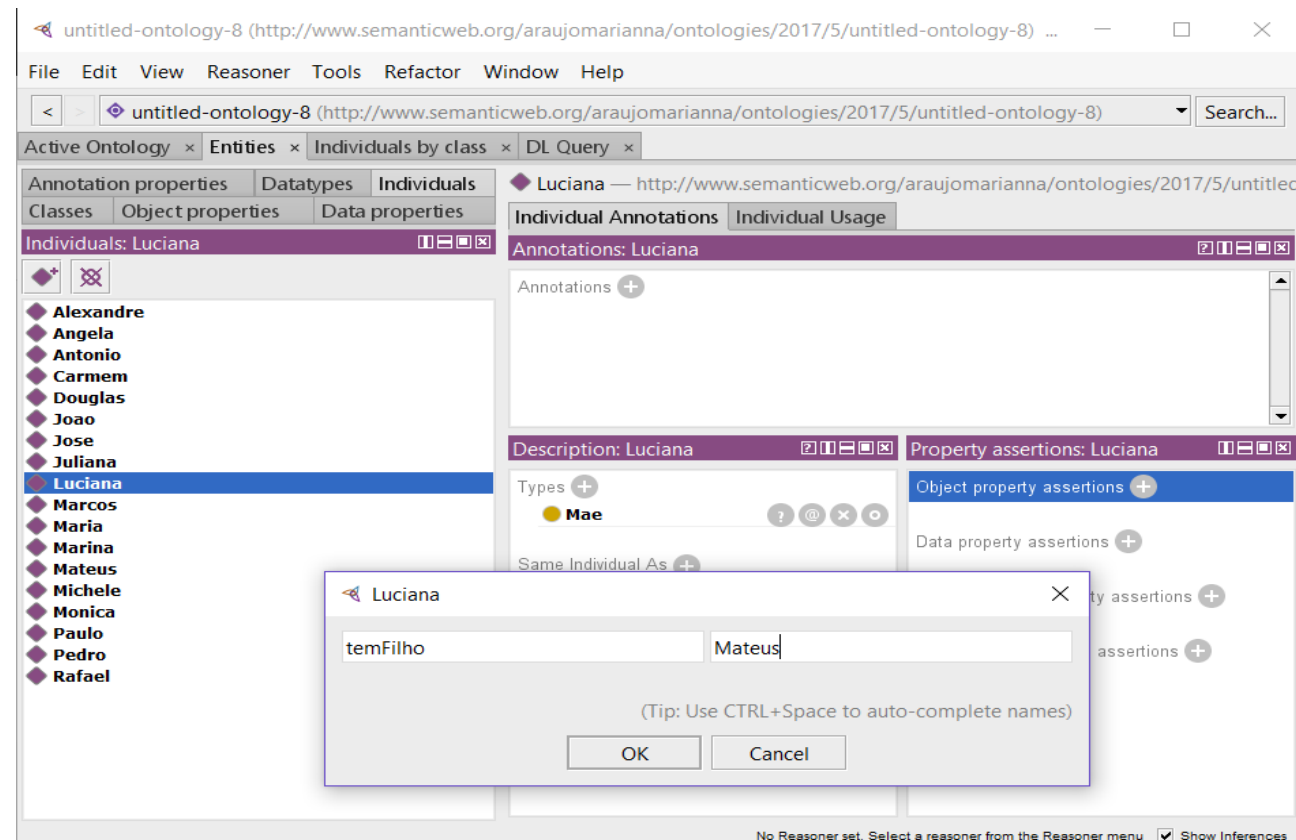
Modelagem de uma ontologia de relações familiares

Adicionar os objetos as classes

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the ontology URL: <http://www.semanticweb.org/araujomarianna/ontologies/2017/5/untitled-ontology-8>. The main workspace is divided into several panes. On the left, the 'Class hierarchy: Filho' pane shows a tree structure starting from 'owl:Thing', followed by 'Pessoa', then 'Filho_G', and finally 'Filho' (which is highlighted). Other classes in the hierarchy include 'Filha', 'Irmão_G', 'Irmão', 'Genitor', 'Pai', 'Mãe', 'Homem', and 'Mulher'. The right pane is titled 'Filho' and contains tabs for 'Class Annotations' and 'Class Usage'. The 'Class Annotations' tab is active, showing a list of annotations for the 'Filho' class. Below this, the 'Description: Filho' pane shows the class's properties, including 'Equivalent To', 'SubClass Of' (with 'Filho_G' listed), 'General class axioms', and 'SubClass Of (Anonymous Ancestor)'. At the bottom, the 'Instances' section lists three individuals: 'Antonio', 'Jose', and 'Rafael'. The status bar at the bottom indicates 'No Reasoner set. Select a reasoner from the Reasoner menu' and has a checkbox for 'Show Inferences' which is checked.

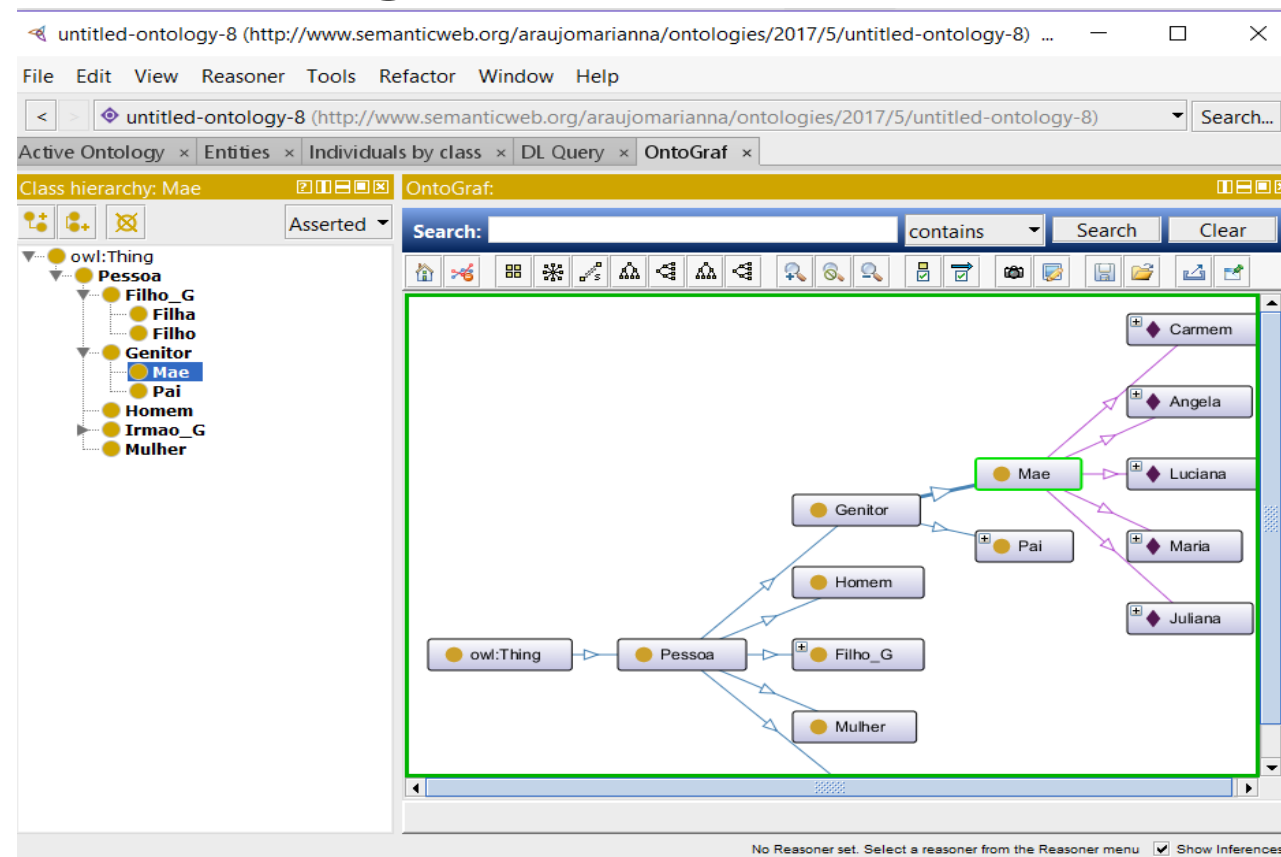
Modelagem de uma ontologia de relações familiares

Relacionando os objetos as classes



Modelagem de uma ontologia de relações familiares

Resultado gráfico da ontologia criada



Referências

Alatrash, E., "Using Web Tools for Constructing an Ontology of Different Natural Languages", Doctoral dissertation, University of Belgrade, 2013

BORST, W.N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. 1997. Tese (Phd). Disponível em: <<http://www.ub.utwente.nl/webdocs/inf/1/t0000004.pdf>>. Acesso em: 10 Junho 2017.

Gruber, T. R., Toward Principles for Design of Ontologies Used for Knowledge Sharing, Formal Ontology in Conceptual Analysis and Knowledge Representation, Guarino, Nicola e Poli, Roberto, editors, Kluwer Academic Publishers, 1993.

Referências

Lima, J. C. and de Carvalho, C. L. (2005). Ontologias - owl (web ontology language). Technical Report - RT-INF-004-05 - Relatório Técnico, Instituto de Informática Universidade Federal de Goiás.

Studer R, Benjamins VR, Fensel D. Knowledge Engineering: Principles and Methods. IEEE Transactions on Data and Knowledge Engineering 25(1-2):161–197.