

Engenharia de Ontologias (Ontology Engineering)

Universidade Federal de Uberlândia

Faculdade de Computação

Programa de Pós-Graduação em Ciência da
Computação

Prof. Fabiano Azevedo Dorça

A disciplina

- Título: "Engenharia de Ontologias e Aplicações Semânticas"
- Ementa: Representação de Conhecimento; Tipos de Ontologia; Linguagem de ontologias da Web; Engenharia de Ontologias; Metodologias de Desenvolvimento de Ontologias; Ontologias em RDF (Resource Description Framework); Manipulação de dados RDF; Ontologias na Web Semântica; Desenvolvimento de Aplicações Semânticas.

A disciplina

- Objetivos:
 - Apresentar uma visão completa sobre ontologias e sua importância na representação de conhecimento e no desenvolvimento de aplicações semânticas.
 - Fornecer técnicas para engenharia de ontologias.
 - Trabalhar com metodologias e ferramentas para desenvolvimento de ontologias.
 - Apresentar a tecnologia RDF e sua importância na representação e manipulação de ontologias na web semântica.
 - Discutir conceitos principais relacionados ao desenvolvimento de aplicações semânticas, padrões de desenvolvimento, ferramentas e plataformas.
 - Possibilitar o desenvolvimento de uma aplicação semântica baseada em ontologias.

A disciplina

- Avaliação
 - Práticas semanais (50% da nota) – em individual ou em dupla. Devem ser entregues (via email) em até 2 dias.
 - Trabalho final – grupo máximo 04 alunos (50% da nota)

Introdução

- Em seu sentido filosófico, **ontologia** é o ramo da metafísica que estuda os **tipos de coisas** que existem no mundo.
- A palavra é derivada do grego ontos, ser, e logos, palavra.
- Seu termo de origem é a palavra aristotélica **categoria**, termo utilizado no sentido de **classificação**.
- Parte da filosofia que trata da **natureza do ser**, ou seja, da realidade, da existência dos entes e das questões metafísicas em geral.

Introdução

- Ontologia em computação é uma técnica de **representação de conhecimento**.
- Criadas por especialistas, tendo sua estrutura baseada na descrição de **conceitos** e dos **relacionamentos** semânticos entre eles.
- Geram uma **especificação formal** e explícita de uma conceitualização **compartilhada**.
- Permite **compartilhamento** e entendimento comum de algum **domínio de conhecimento** que possa ser comunicado entre pessoas e computadores.

Introdução

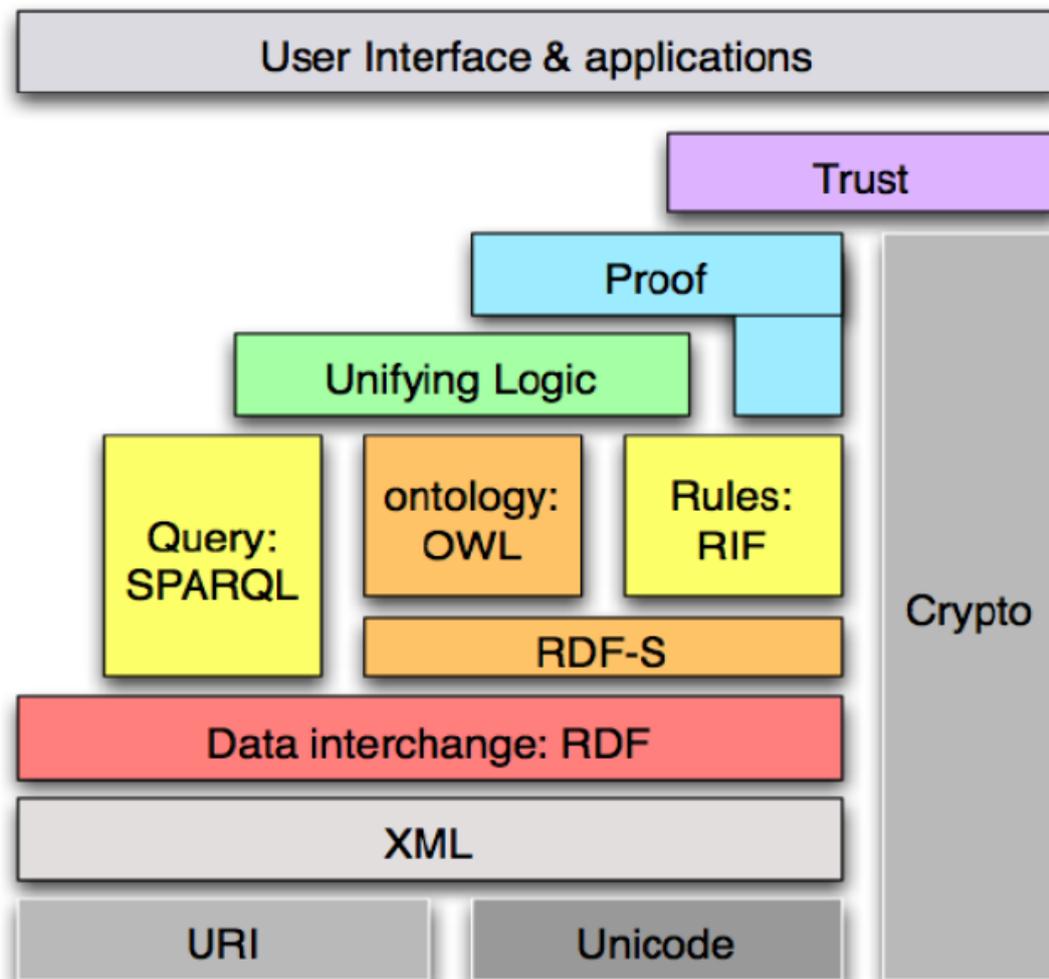
- Possibilita o **compartilhamento e reutilização** de conhecimento.
- Permite que máquinas processem e integrem recursos de Internet de maneira **inteligente**.
-
- Permite **raciocínio** sobre o conhecimento (reasoning).
- Possibilita **comunicação** entre dispositivos heterogêneos na rede.
- São representadas em **lógicas de descrição** (DLs), por exemplo, **lógica de primeira ordem** (cálculo de predicados).

Introdução

- A **linguagem de ontologia** mais amplamente utilizada para fins computacionais, é a *Web Ontology Language (OWL)*.
- Primeira versão surgiu em 2004 e uma 2a. versão em 2009 (OWL2).
- Tem sido o **padrão** de desenvolvimento de sistemas e **aplicações semânticas** baseadas ontologias.

Introdução

- OWL não existe isoladamente, mas é parte da pilha da Web Semântica.



Introdução

- Web Semântica
 - informação deve ter significado (semântica) explícito
 - máquinas possam processá-la de forma mais inteligente.
- XML – permite definir sintaxe padrão para conceitos, mas não a semântica dos mesmos
- OWL – permite representar a semântica das coisas, representando recursos web através de uma DL.
- Máquinas podem usar algoritmos de **inferência** para **raciocinar** sobre os termos.

eXtensible Markup Language

- Na **parte inferior da pilha**, temos XML, que não tem semântica, mas permite a descrição estruturada de **dados**.
- Não permite descrição de conhecimento e raciocínio.
- XML é uma **linguagem de marcação** que permite a criação e o uso de *tags* personalizadas.
- Permite uma maneira simples de **organizar e estruturar** os dados.

eXtensible Markup Language

- Possibilita **comunicação** entre aplicações (troca de dados padronizados).
- Muito usada em protocolos de **comunicação** em **sistemas distribuídos**.
- Atualmente XML é a **linguagem padrão** recomendada pela **W3C** para troca de dados via Web.

eXtensible Markup Language

- Sintaxe do XML é bem **formal** e possui uma série de **regras** a serem obedecidas.
- Um documento XML que segue **todas as regras** sintáticas da linguagem é chamado de documento **XML bem formado**.

eXtensible Markup Language

- As **tags** XML são usadas para armazenar e organizar os dados
- Existem **três características importantes de XML** que a tornam útil em uma variedade de sistemas e soluções:
 - XML é **extensível**: o XML permite criar *tags* próprias.
 - XML **carrega** os dados, não apresenta: XML permite armazenar dados independentemente de como será apresentada.
 - XML é um **padrão público**: XML foi desenvolvido por uma organização chamada World Wide Web Consortium (W3C) e está disponível como um **padrão aberto**.

eXtensible Markup Language

- XML é uma linguagem de marcação
 - Markup (marcação) usadas para demarcar e rotular os dados.
- Exemplo:

```
<message>  
  <text>Hello, world!</text>  
</message>
```

–

- As tags `<message>` e `</ message>` marcam o início e o final do fragmento de código XML.

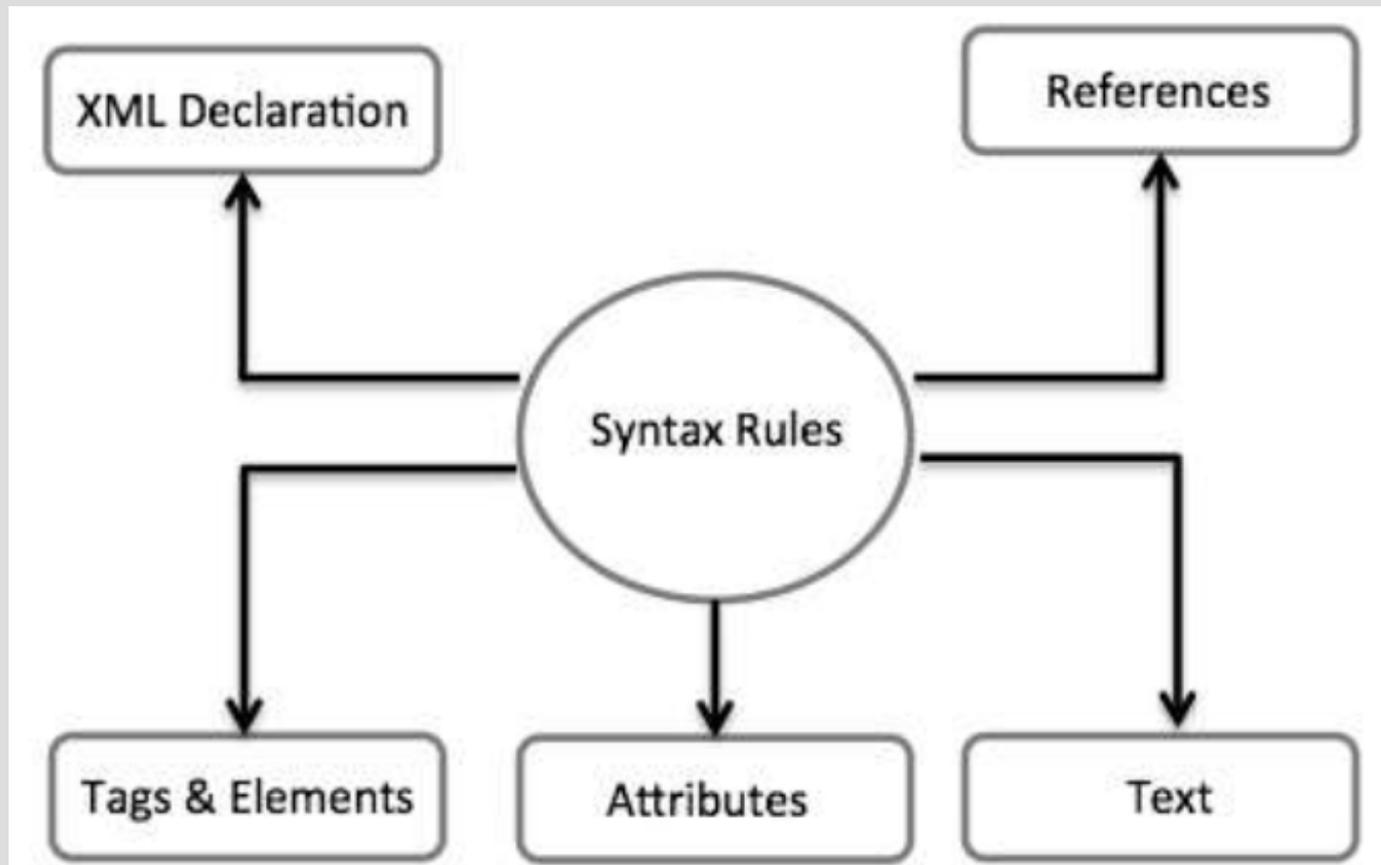
eXtensible Markup Language

- Segue-se um documento XML completo:

```
<?xml version="1.0"?>
<contact-info>
  <name>João</name>
  <company>UFU</company>
  <phone>(034) 123-4567</phone>
</contact-info>
```

eXtensible Markup Language

- Partes de um documento XML



eXtensible Markup Language

- XML Declaration
 - `<?xml version="1.0" encoding="UTF-8"?>`
 - Onde *version* é a versão XML e *encoding* especifica a codificação de caracteres usada no documento.
 - UTF-8 é a codificação default, assumida quando não se explicita na declaração.
 - Documentos XML são case sensitive.

eXtensible Markup Language

- Tags e Elementos
 - Um arquivo XML é estruturado por vários **elementos** XML, também chamados **XML-nodes** ou XML tags.
 - Os nomes dos **elementos** XML estão entre parênteses triangulares <>, como mostrado abaixo:
 - <element>
- Cada elemento XML precisa ser **fechado**
 - <element>....</element>
 - <element/> (elemento vazio)

eXtensible Markup Language

- Elemento raiz
 - Um documento XML pode ter **apenas 1 elemento raiz**.
 - Errado - ambos os elementos x e y ocorrem no nível superior, sem um elemento raiz:
 - `<x> ... </ x>`
 - `<y> ... </ y>`
 - Correto - documento XML formado corretamente:

```
<root>
  <x> ... </ x>
  <y> ... </ y>
</ root>
```

eXtensible Markup Language

- Elementos XML devem ser devidamente aninhados
- Em HTML, você pode ver elementos aninhados **incorretamente**:
 - ` <i> Este texto é negrito e itálico </i>`
- Em XML, todos os elementos devem estar **corretamente** aninhados entre si:
 - ` <i> Este texto é negrito e itálico </i> `

eXtensible Markup Language

- Conteúdo dos elementos
 - Os elementos podem conter texto, outros elementos ou estarem vazios.
 - Os exemplos de elementos HTML vazios são `<hr/>`, `
`.

eXtensible Markup Language

- Atributos XML
 - Um atributo especifica uma única **propriedade** para o **elemento**, usando um **par nome / valor**.
 - Um XML element pode ter zero ou mais atributos. Por exemplo:
 - `UFU`
 - ``

eXtensible Markup Language

XML References

- Elementos especiais em um documento XML.
- As referências começam sempre com o símbolo "&", que é um caractere reservado e termina com o símbolo ";".
- Exemplo:
 - - Alguns caracteres têm um significado especial em XML.

eXtensible Markup Language

- Colocar um caractere como "<" dentro de um elemento XML, ele gerará um erro porque o analisador interpretá-lo como o início de um novo elemento.
- Isso gerará um erro XML:
- `<message> salário <1000 </message>`
- Para evitar esse erro, substitua o caractere "<" por uma referência de entidade:
- `<message> salary < 1000 </message>`

eXtensible Markup Language

Not Allowed Character	Replacement Entity	Character Description
<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

eXtensible Markup Language

- Texto em XML
- UTF-8 é a codificação *default* para texto XML
 - oferece o armazenamento em um 1 byte para os caracteres mais comuns e 2 bytes para caracteres especiais (**tamanho variável**)
 - pode representar qualquer caracter universal padrão do **Unicode**, sendo também compatível com o **ASCII**.
-

eXtensible Markup Language

- A codificação ISO-8859-1 **sempre utiliza um 1 byte** por caractere, e,
 - por isso, é bem mais **limitada** que a UTF-8.
 - a codificação ISO-8859-1 não possui as aspas angulares: “ (abre aspas) e ” (fecha aspas).
- Seria necessário utilizar **entidades ou referências numéricas** para codificá-las em um documento XML caso fosse utilizada a codificação ISO-8859-1.
- Outra vantagem da UTF-8 é a possibilidade de armazenamento de documentos textuais de outros países que utilizam conjuntos de caracteres não latinos.

eXtensible Markup Language

- Exemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!-- Comentário... -->
```

```
<note>
```

```
  <to>João</to>
```

```
  <from>Maria</from>
```

```
  <heading>Reunião</heading>
```

```
  <body>Reunião agendada para amanhã as 14hs.</body>
```

```
</note>
```

eXtensible Markup Language

- Regras de Comentários XML
 - Os comentários não podem aparecer antes da declaração XML.
 - Os comentários podem aparecer em qualquer parte de um documento.
 - Os comentários não podem aparecer em valores de atributos.
 - Os comentários não podem ser aninhados dentro dos outros comentários.

eXtensible Markup Language

- Doc. XML bem-formatado x Válido
 - Documento XML bem-formatado:
 - documento que segue as regras de formação do XML (raiz única, bem aninhado, etc.)
 - Documento XML válido:
 - documento bem-formatado que segue as regras de um **esquema** associado

eXtensible Markup Language

- Tipos de esquema
 - DTD - Document Type Definition
 - Gramática Regular
 - XML Schema
 - Esquema escrito em XML
- Se um documento XML for bem-formado e obedecer a uma DTD ou XML Schema associado, então é dito ser um documento XML válido.

eXtensible Markup Language

- OK!