

---

# Representação do Conhecimento: Representação da Informação

José Carlos Ramalho  
[jcr@di.uminho.pt](mailto:jcr@di.uminho.pt)

baseado em tutoriais e apresentações: Sean Bechhofer, Vagan Terziyan,  
Holger Knublauch, José Carlos Ramalho

Outubro de 2008

---

# A Disciplina

---

- 2 sessões de 4h + 1 sessão de 1h
- 1ª parte: Linguagens de Anotação
- 2ª parte: RC na Web, Ontologias
- 3ª parte: ?
- Avaliação:
  - 1 trabalho de grupo: desenvolvimento de uma ontologia
  - 1 trabalho individual escrito

# Trabalho Individual

---

- Desenvolver um dos temas, em formato de artigo, com um máximo de 7 páginas;
- Temas disponíveis:
  - Web 2.0, OWL, RSS, Wikis, RDF, XBRL, Topic Maps, GML, Web Semântica, Ontologias para a Web.
- Estrutura convencional:
  - Front-matter: identificação, resumo, palavras-chave, etc.
  - Body: secções, subsecções, parágrafos, imagens, etc.
  - Back-matter: referências/bibliografia, anexos.

# Considerações

---

- RC é inicialmente interessante;
- A sua utilização é feita em pequena escala nunca atingiu os níveis de utilização pretendidos;
- Este era o estado do Hipertexto antes do aparecimento da Web;
- RC pode tornar-se a Web Semântica de amanhã se sofrer o mesmo processo de globalização;
- Removem-se os conceitos de verdade absoluta, conhecimento total e vê-se o que se consegue com conhecimento limitado.



# Premissa

---

Globalização da Representação do  
Conhecimento

=~

Web Semântica

# Três vias alternativas para o desenvolvimento Web

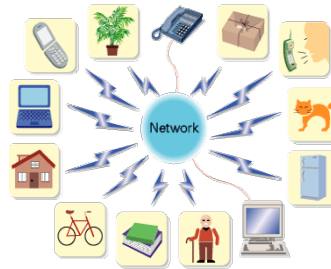


Comunidades

Web 2.0

- Wikis
- Blogs
- Mashups
- Portais

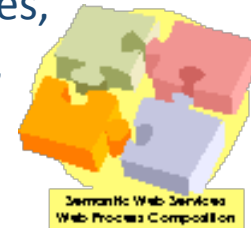
Equipamento



Web of Things

- Computação Ubíqua
- Espaços Inteligentes
- RFID
- Sistemas Embebidos
- Redes Sensoriais

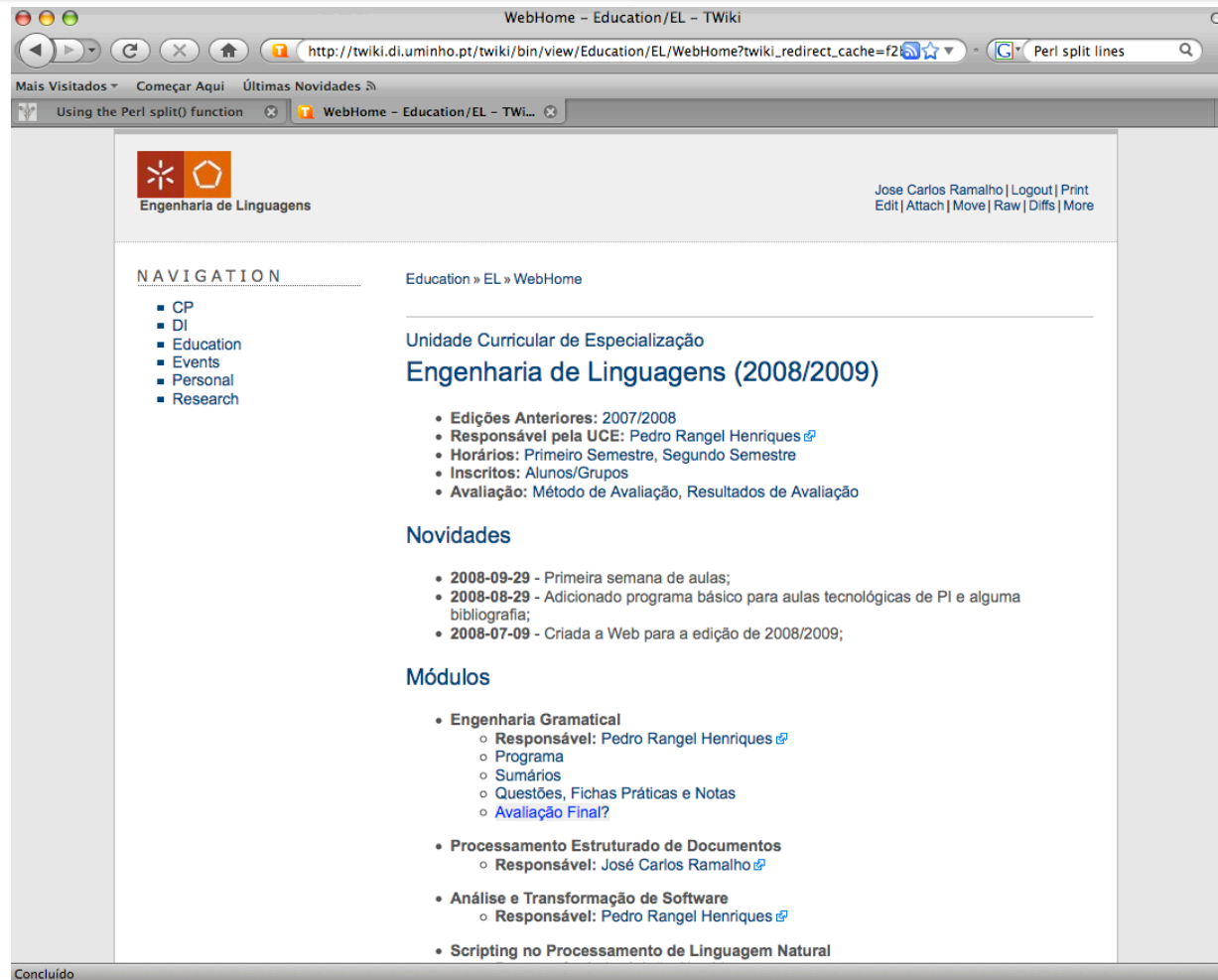
Aplicações,  
Serviços,  
Agentes



Semantic Web

- Metainformação
- Ontologias
- Serviços Web
- Agentes
- Portais Inteligentes

# Conceito: Wiki



# Conceito: Blog

The image shows a screenshot of a web browser with two tabs. The left tab is titled 'mis dibujos' and displays a blog post. The right tab is titled 'action' and displays a video player with a YouTube logo.

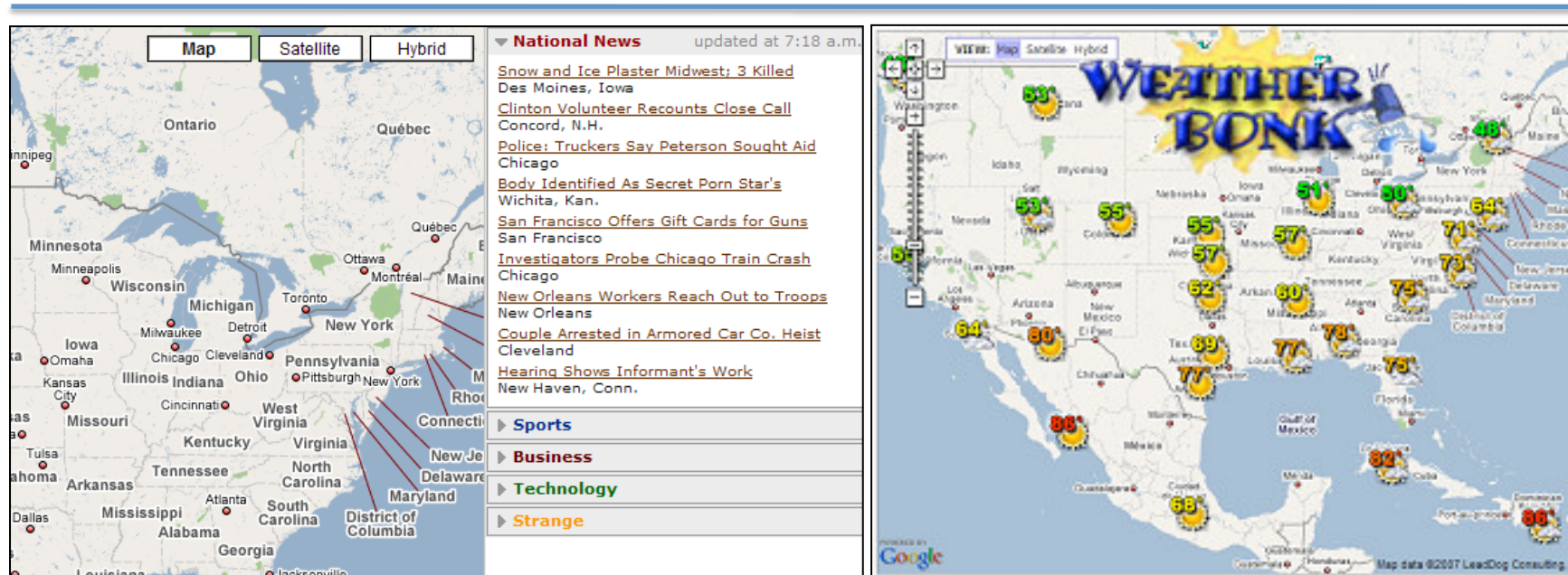
**Blog Post (Left Tab):**

- Navigation: PESQUISAR NO BLOGUE | ASSINALAR BLOGUE | Blogue seguinte»
- Header: **Kristi's adventures**
- Date: **friday, october 17, 2008**
- Post Title: **Ugh!**
- Text: I have so much to post but having trouble with my internet connection!
- Posted by: kristi at 5:50 AM
- Post Title: **Red Chili**
- Blog Archive: 2008 (68) | October (63) | [Ugh!](#) | [Red Chili](#) | [Aloha from Uganda!](#) | [Lions & tigers & elephants, oh my!](#) | [Jazz Festival!](#) | [Out to Garden City](#) | [Safari](#) | [This is Kampala](#) | [Makada](#) | [The premies](#) | [Alvin](#) | [Amos](#) | [My girls](#) | [Amalie](#) | [Me & Grace](#) | [Peace's story](#) | [My new room](#) | [The quest house](#) | [What Now](#) | [More medical stuff](#) | [Vicky](#) | [Me & Izzy](#) | [Church at KPC](#)
- Images: A green tent in a field and a wooden structure in a rural setting.

**Video Player (Right Tab):**

- Video Title: **mi mario del alma , este man es un bien. lastima que lo hagan enojar tanto**
- Video Content: A man speaking at a podium.
- Channel: realizado por **YouTube**
- Blog Archive: ARCHIVO DEL BLOG | 2008 (11) | octubre (11)
- Text: por ahora se me acabaron las fotos, un dia de esto... mi mario del alma , este man es un bien. lastima... flabio que mona mas linda a esta gasela no la alcanza nadie. que mar... pa... se me colo este pato. no podia faltar , en todas ... la pareja del siglo no son una belleza?a lo bie... quien sera?. que nos molesta y nos jode la vida ... don ruben: con todo y su pecho de palomojajajaja....

# Mashups



Mashup é um termo que designa a combinação de funcionalidades de vários sistemas acessíveis via Web.

# Semantic Web

---

- A Web tornou-se possível devido a algumas normas
  - **TCP/IP** para o transporte de bits através de redes
  - **HTTP & HTML** para transportar e formatar conteúdos em formato hipertexto
- e a aplicações capazes de explorar este cenário
- **1ª geração:** páginas HTML escritas à mão
- **2ª geração:** páginas geradas por máquinas
  - dirigidas ao processamento e interacção humanos
- Na próxima geração, os recursos devem estar acessíveis a processos automáticos

# História da Web Semântica

---

- A Web foi “criada” por Tim Berners-Lee (entre outros), quando trabalhava no CERN
- A sua visão original era mais ambiciosa que a realidade actual:

...a goal of the Web was that, if the interaction between person and hypertext could be so intuitive that the machine-readable information space gave an accurate representation of the state of people's thoughts, interactions, and work patterns, then machine analysis could become a very powerful management tool...



Muitos têm trabalhado para realizar esta visão que se tornou conhecida como a “Web Semântica”.



# Hoje: a Web Sintáctica

**José Carlos Ramalho**  
Prof. Auxiliar  
Department of Informatics // CCTC Research Center  
University of Minho  
Campus de Gualtar 4700-057 Braga - Portugal  
Phone: +351 253 604479 | FAX: +351 253 604471 | Ext.: 604479  
jcr @ di.uminho.pt | http://www.di.uminho.pt/~jcr/

[Personal information](#) | [Research projects](#) | [Publications](#) | [Communications](#) | [Thesis Supervision](#) | [Thesis Examination](#) | [Conferences](#) | [Teaching](#) | [Personal hobbies](#)

**Personal information**

José Carlos Ramalho graduated in 1991 as a Systems and Informatics Engineer. During his graduation he has worked as a freelancer software developer and as a network technician for Apple. During this time he also taught several courses about computer programming for several public institutions.

In 1991 he joined the Department of Informatics as an Assistant and started his MSc work. He has worked as a teacher since 1991 until the present. He also has been a researcher of Algorithm's Research Center until 2004. In 2004 he moved from Algorithm's to the newly created CCTC Research Center.

He finishes his PhD in 2000 under the subject "Structured Documents Semantics". From 1996 until the present he has been researching in Structured Documents area (currently he is coordinating several projects in Digital Archives and Libraries field). He was responsible for the creation of the conference series called XATA (starting in 2003 until now). He wrote two books and many articles presented in international and national conferences.

**Academic background**

(2000-1996) Phd: "Anotação Estrutural de Documentos e sua Semântica" at University of Minho.

(1993-1991) Msc: "Um Compilador para o GLITCH" at University of Minho.

(1991-1985) Graduation: Systems and Informatics Engineering, at University of Minho.

**Personal hobbies**

**My Sax Adventure**

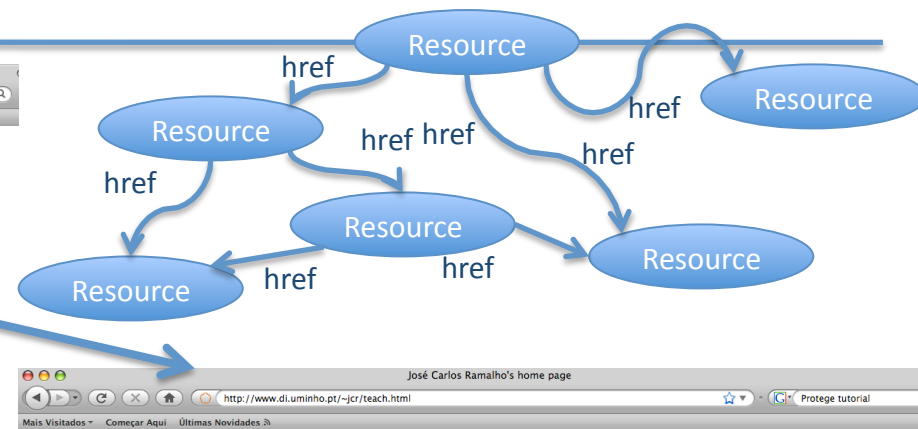
**Squash**

[World Squash](#) | [Guide to squash](#)

**Geocaching**

[Geocaching - The Official Global GPS Cache Hunt Site](#)

Found: 73 / Hidden: 12  
*Adventure for the whole family!*



**José Carlos Ramalho**  
Prof. Auxiliar  
Department of Informatics // CCTC Research Center  
University of Minho  
Campus de Gualtar 4700-057 Braga - Portugal  
Phone: +351 253 604479 | FAX: +351 253 604471 | Ext.: 604479  
jcr @ di.uminho.pt | http://www.di.uminho.pt/~jcr/

[Personal information](#) | [Research projects](#) | [Publications](#) | [Communications](#) | [Thesis Supervision](#) | [Thesis Examination](#) | [Conferences](#) | [Teaching](#) | [Personal hobbies](#)

**Courses**

**Processamento de Linguagens I**  
2000; 2001, 2002, 2005.

**Processamento de Linguagens e Compiladores**  
2008.

**Processamento Estruturado de Documentos**  
2000; 2001, MICEI02, 2002/2003, MICEI03, 2003/2004, MICEI04, 2004/2005, MICEI2005, PED2005, PED2007 (UCE - Engenharia de Linguagens).

**Linguagens de Anotação de Documentos**  
2003, 2004.

**Programação Imperativa**  
2005, 2006, 2008.

**Programação Imperativa / Paradigmas da Programação II**  
2004, 2006.

**Paradigmas da Programação I / Programação Funcional**  
2006, 2007.

**Laboratórios de Comunicações III**  
2007/2008, 2008/2009.

- Um lugar onde os computadores fazem a apresentação da informação (fácil) e as pessoas a interpretação e a navegação (difícil).
- Ideia: Porque não colocar o computador a fazer uma maior parte do trabalho difícil?



# Tarefas difíceis na Web Sintáctica...

---

- Queries complexas envolvendo **conhecimento anterior**
  - Encontra informação sobre “animais que usam radar mas não são morcegos, nem golfinhos nem baleias”

- Localização de informação em repositórios

– Pesquisa de viagens

Quase impossível para as máquinas e muito difícil para as pessoas sem nenhum grau de automação

- Delegação em agentes web
    - Reserva-me umas férias no próximo fim-de-semana nalgum lugar quente, não muito longe, e onde se fale inglês ou francês
-

# Qual é o problema?



Uma página Web é composta por:

- informação de formatação (e.g. tamanho de letra, cor);
- links para conteúdos relacionados.

O Conteúdo semântico é acessível aos humanos mas não aos computadores (pelo menos de um modo fácil).

# A informação que vemos...

---

- **iPRES 2008: The Fifth International Conference on Preservation of Digital Objects**
  - The British Library will host this year's International Conference on Preservation of Digital Objects (iPRES 2008) at its Conference Centre in St Pancras, London, on 29-30 September 2008.
- **Program**
  - iPres 2008 will feature two tracks to cater for the wide ranging interests of iPres delegates...

# A informação que as máquinas vêm...

---

0000040	042057	072141	037145	020012	020040	036040	060503	060564
0000060	067554	072547	051545	060564	072564	037163	071104	063141
0000100	036164	041457	072141	066141	063557	062565	072123	072141
0000120	071565	005076	020040	020040	041474	062562	072141	062145
0000140	030076	027466	027470	034461	034471	027474	071103	060545
0000160	062564	037144	020012	020040	036040	062522	067560	064563
0000200	067564	074562	040476	052116	036124	051057	070145	071557
0000220	072151	071157	037171	020012	020040	036040	064524	066164
0000240	037145	064103	067141	062543	060554	064562	020141	062544
0000260	042040	020056	060515	064562	020141	036111	052057	072151
0000300	062554	005076	020040	020040	042474	072170	067145	037164
0000320	034464	066040	073151	036056	042457	072170	067145	037164
0000340	020012	020040	036040	067515	064544	064546	062145	030076
0000360	027467	033061	031057	030060	036070	046457	062157	063151
0000400	062551	037144	020012	020040	036040	062522	047146	037157
0000420	052120	052055	026524	027501	027506	027501	036062	051057
0000440	063145	067516	005076	020040	020040	047074	072157	071545
0000460	047076	072157	020141	067541	062440	062554	062555	072156
0000500	020157	062544	064440	063156	071157	060555	123703	121703
0000520	020157	061442	131703	064544	067547	062040	020145	062562
0000540	062546	141562	067252	064543	021141	020072	141543	062263
0000560	063551	020157	062544	071040	063145	071145	125303	061556
0000600	060551	060440	072156	063551	035157	050040	026524	052124
0000620	046455	047503	041457	046517	041457	041517	031057	036056

# Limitações actuais da Web

---



Interacção Homem-Máquina



Interacção Máquina-Máquina

# As Máquinas não entendem o Significado

---

- “*O meu rato está partido. Preciso de um novo...*”

## Uso de Ontologias

“O meu rato está *partido*”  
vs. “O meu rato está *morto*”



# É preciso adicionar “Semântica”

---

- Metainformação: informação sobre a informação.
- Acordo no significado das anotações.
  - Dublin Core na anotação de conteúdos bibliográficos
- Uso de Ontologias
  - É a especificação de uma conceptualização;
  - É uma descrição dos conceitos e relações que podem existir para um agente ou uma comunidade de agentes.

# Como construir uma Web Semântica

---

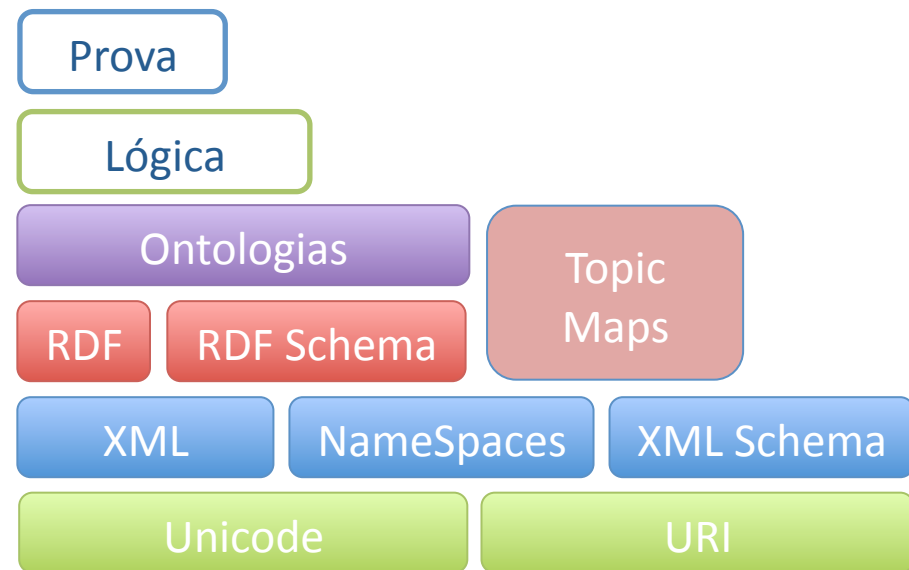
- Anotando
  - Associando metainformação aos recursos
- Integrando
  - Integrando os vários recursos de informação
- Inferindo
  - Raciocinando sobre a informação que se possui;
  - Pode ser “leve”(taxonomia)
  - Pode ser “pesada” (lógica)
- Interoperabilidade e Partilha são também chave



# Linguagens

---

- O trabalho nesta área tem-se focado na definição de uma pilha de linguagens.
- Estas linguagens são usadas para suportar a representação e o uso de metainformação



---

# Linguagens de Anotação:

## XML - Aplicações e Tecnologias Associadas

# Motivação 1

---

- Um dia de trabalho = produção de vários documentos
- Muitos têm uma forma estruturada
- Alguns até podem ser representados numa tabela: inventários, preços, ...
- Mas, apenas 10% da informação é guardada em Bases de Dados
- Que fazer com os restantes 90%?

# Os restantes 90%

---

- Correspondem a textos que circulam dentro das instituições
- Não se lhes pode aplicar uma metodologia relacional
- Haverá alguma maneira de contornar o problema?
  
- A solução recai sobre a estruturação da informação

# Exemplo: uma carta

---

Exmo Vice-Reitor  
Prof. Dr. José Viriato Eiras Capela

Devido à proximidade do prazo e ao trabalho em que ... venho, por este meio, solicitar-lhe que conceda mais 30 dias para a submissão final da tese de mestrado a dois dos meus orientandos: Joel Vicente (Mestrado em Informática) e Luis Miguel Alves Domingues (Mestrado em Informática).

...

Com os melhores cumprimentos

Universidade do Minho, Braga,  
21 de Setembro de 2005

José Carlos Leite Ramalho  
(Professor Auxiliar)

# Motivação 2

---

- Publicação Electrónica
  - Proliferação das TICs = proliferação de formatos
  - Explosão da Web veio agravar ainda mais
  - Questão: Como conseguir produzir documentos num formato neutro a partir do qual seja possível gerar todos os formatos necessários para distribuição?

**Será possível atingir a interoperabilidade sem um denominador comum?**

# Documentação Estruturada

---

- Valor de um documento = facilidade na localização, no consumo, na validação e na reutilização
- Um documento estruturado tem as seguintes vantagens:
  - Acesso
  - Validação
  - Reutilização
  - Normalização

# Anotação

---

- “Markup” = anotação, codificação, etiquetagem
- A anotação de um texto é um meio de tornar explícita uma interpretação desse texto
- Exemplo:
  - “Está a chover.”
  - “Está a chover?”



# Objectivos da Anotação

---

1. Dividir o documento em componentes
  - Dá organização lógica (explicitamente)
  - Dá indicações para o processamento (implicitamente)
2. Associar semântica
  - Dá interpretação (implicitamente)
  - Dá indicações para a formatação (explicitamente)

# Fases da Anotação

---

1. Análise da estrutura da informação (dos documentos que se pretende tratar).
2. Definição da formatação/transformação desejada para cada elemento estrutural.
3. Inserção das anotações no documento.

# Pausa para pensar

---

- Exercício: anotar o poema
- Exercício: definir as anotações para a agenda
- Exercício: anotar um relatório

# Exercício: o poema

"Soneto Já Antigo"  
(Álvaro de Campos)

Olha, Daisy: quando eu morrer tu hás-de  
dizer aos meus amigos aí de Londres,  
embora não o sintas, que tu escondes  
a grande dor da minha morte. Irás de

Londres p'ra Iorque, onde nasceste (dizes  
que eu nada que tu digas acredito),  
contar áquele pobre rapazito  
que me deu horas tão felizes,

embora não o saibas, que morri...  
Mesmo ele, a quem eu tanto julguei amar,  
nada se importará... Depois vai dar

a notícia a essa estranha Cecily  
que acreditava que eu seria grande...  
Raios partam a vida e quem lá ande!

(1922)

**Poema:** título, autor, corpo, data

**Corpo:** quadra, quadra, terceto,  
terceto.

**Quadra:** verso, verso, verso,  
verso

**Terceto:** verso, verso, verso

**Verso:** (texto | nome)+

**Nome:** texto

# Anotação Descritiva

---

```
<carta>
<destinatario> Exmo Vice-Reitor
Prof. Dr. José Viriato Eiras Capela </destinatario>
<corpo>
Devido à proximidade do prazo e ao trabalho em que ... venho, por este
meio, solicitar-lhe que conceda
mais 30 dias para a submissão final da tese de mestrado a dois dos meus
orientandos: Joel Vicente (Mestrado em Informática) e Luis Miguel Alves
Domingues (Mestrado em Informática).
... </corpo>

<fecho> Com os melhores cumprimentos
Universidade do Minho, Braga,
21 de Setembro de 2005
José Carlos Leite Ramalho
(Professor Auxiliar) </fecho>
</carta>
```

**Utiliza etiquetas para apenas classificar as componentes do documento.**

# Linguagem de Anotação

---

- Especifica como distinguir a anotação do texto
- Especifica **que** anotações são **necessárias** e **quais** são **permitidas**
- Especifica **onde** as anotações são **neces-sárias** e **onde** são **permitidas**
- Define o significado da anotação

O XML tem estas características todas ... excepto a última

# Perspectivas de Anotação

---

1. Anotação orientada ao formato
  2. Anotação orientada à estrutura
  3. Anotação orientada ao conteúdo
- Objectivo: Anotação Equilibrada

# Anotação orientada ao formato

---

**<quadra>**

Olha, **<realçado>** Daisy **</realçado>** : quando eu morrer tu hás-de dizer aos meus amigos aí de **<realçado>** Londres **</realçado>**, embora não o sintas, que tu escondes a grande dor da minha morte. Irás de **</quadra>**



# Anotação orientada à estrutura

---

```
<SEC>Isto é uma secção de nível 1.  
<SEC>Isto é uma secção de nível 2.</SEC> </SEC>  
<P>Isto é um parágrafo do nível de topo.</P>  
<LISTA>  
  <ITEM>Isto é um item de uma lista de nível 1.  
  <LISTA><ITEM>Isto é um item de uma lista de  
  nível 2.</ITEM>  
</LISTA></ITEM></LISTA>
```

```
<SEC1>Isto é uma secção de nível 1.</SEC1>  
<SEC2>Isto é uma secção de nível 2.</SEC2>  
<P0>Isto é um parágrafo do nível de topo.</P0>  
<LISTA1><ITEM>Isto é um item de uma lista de nível 1.</ITEM></LISTA1>  
<LISTA2><ITEM>Isto é um item de uma lista de nível 2.</ITEM></LISTA2>
```

# Anotação orientada ao conteúdo

---

...

**<quadra>**

**<verso>Olha, <nome>Daisy</nome>: quando eu morrer  
tu hás-de</verso>**

**<verso>dizer aos meus amigos aí de  
<lugar>Londres</lugar>,</verso>**

**<verso>embora não o sintas, que tu escondes</verso>**

**<verso>a grande dor da minha morte. Irás de</verso>**

**</quadra>**

...

# Anotação Equilibrada

---

- Exemplo: DocBook
  - Formato: EMPH, TABLE,...
  - Estrutura: SECT1, SECT2, SECT3, ...
    - Conteúdo: NAME, AUTHOR, PUBDATE, COMMAND, ...

---

# Documentos XML bem formados

# Um documento XML

---

- Conteúdo = Dados + Anotações
- Dados = blocos de texto
- Anotações:
  - marcas de início de elementos
  - marcas de fim de elementos
  - marcas de elementos vazios
  - referências a entidades
  - comentários
  - limitadores de secções especiais de texto
  - declarações de tipo de documento
  - instruções de processamento

# O exemplo tradicional

---

```
<?xml version="1.0" encoding="UTF-8"?>  
<doc>  
    Hello World!!!  
</doc>
```

# A declaração XML

---

- Anotação especial que deve iniciar todos os documentos XML

```
<?xml  
  version="1.0"  
  standalone="yes"  
  encoding="utf-8" />
```

**version** - obrigatório, valores possíveis: 1.0  
**standalone** - opcional, valores possíveis: yes, no;  
**encoding** - opcional, para o português o valor deverá ser: ISO-8859-1

# Comentários

---

- Podem aparecer em qualquer ponto dum documento XML.
- Começam pela marca: <!--
- e terminam com a marca: -->.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!--Isto é um comentário no início-->
<doc>
    Olá Mundo!!!
</doc>
```



# Instruções de Processamento

---

- As instruções de processamento são uma reminiscência da anotação procedimental
- Uma instrução de processamento não faz parte do conteúdo do documento.
- É uma indicação directa de que algo deve ser executado naquele ponto.
- Uma instrução de processamento começa por: `<?id-processor`
- e termina por: `?>`
- Exemplo: a declaração XML

# Instruções de Processamento 2

---

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<agenda>
  <?html action="hr"?>
    <entrada id="e1" tipo="pessoa">
      <nome>José Carlos Ramalho</nome>
      <email>jcr@di.uminho.pt</email>
      <telefone>253 604479</telefone>
    </entrada>
  <?html action="hr"?>
  ...
</agenda>
```

# Elementos

---

- Blocos lógicos em que um documento pode ser decomposto
- Exemplo:
  - Vais ver o espectáculo a <lugar>Braga</lugar>?
- Uma anotação de início começa por < e termina por > ,
- e uma anotação de fim começa por </ e termina por > .
- Uma anotação contém o nome do elemento que inicia ou que termina, respectivamente.

# Caracteres reservados

---

- No conteúdo dum elemento, nunca deverão ser usados os caracteres '<' e '>' pois são os caracteres que limitam as anotações.
- Em lugar deles devem-se usar, respectivamente, as entidades do tipo carácter '&lt;' e '&gt;'.
- Qualquer processador ou editor de XML fará a substituição automática daquelas entidades pelos caracteres correspondentes.

# Tipos de Conteúdo

---

- elemento com conteúdo textual
  - <lugar>Braga</lugar>
  - <INGREDIENTE>Meia dúzia de ovos</INGREDIENTE>
  - <data>(1922)</data>
- elemento com conteúdo misto
  - <verso>Olha, <nome>Daisy</nome>: quando ...</verso>  
<p>Vais ver o espectáculo a <lugar>Braga</lugar>?</p>

# Tipos de Conteúdo (2)

---

- elementos com conteúdo vazio: são normalmente utilizados pelo seu significado posicional - referências, pontos de inserção de imagens, ...
  - **Como será discutido num capítulo mais à frente (<ref ident="cap5"/>) ...**
  - São representados por uma única anotação que é iniciada por '<' e termina em '/>', que é a forma abreviada de escrever “<elem-ident></elem-ident>”.

# Atributos

---

- Um elemento pode ter um ou mais atributos que, por sua vez, podem ser opcionais ou obrigatórios.
- Visam qualificar o elemento a que estão associados.
- Não há limite para o número de atributos que podem estar associados a um elemento.
- Aparecem sempre na anotação que marca o início dum elemento, uma vez que vão qualificar o conteúdo que se segue.

# Atributos (2)

---

- Um atributo é definido por um par constituído por um nome e um valor:
  - o nome e o valor devem estar separados pelo sinal '=' e
  - o valor deverá estar colocado dentro de aspas simples ou duplas.
  - Exemplo:
    - <ref destino="exemplo5"/>
    - <imagem path="figs/img3.gif"/>



# Elemento versus Atributo

---

- Não existe uma fronteira entre os dois e muitas vezes a escolha não é simples.

## Informação nos elementos

```
<agenda>  
  <entrada id="e1" tipo="pessoa">  
    <nome>José Carlos Ramalho</nome>  
    <email>jcr@di.uminho.pt</email>  
    <telefone>253 604479</telefone>
```

## Informação nos atributos

```
</entrada>  
...  
</agenda>  
  
<agenda>  
  <entrada id="e1" tipo="pessoa" nome="José Carlos Ramalho"  
    email="jcr@di.uminho.pt" telefone="253 604479"/>  
  ...  
</agenda>
```

# Regras de bem-formação

---

- Um documento XML deve ter sempre uma declaração XML no início
- O documento deve incluir um ou mais elementos
- Todos os elementos têm anotações de início e fecho (excepto os vazios)
- Os elementos deverão estar aninhados correctamente
- Os valores de atributos têm de estar dentro de aspas

# Documento XML exemplo

---

```
<?xml version="1.0" encoding="UTF-8"?>
<tree>
  <node id="peter">
    <name>Peter</name>
    <birth>1991</birth>
    <mother refid="mary"/>
    <father refid="joseph"/>
  </node>

  <node id="mary">
    <name>Mary</name>
    <birth>1956</birth>
  </node>
```

```
<node id="joseph">
  <name>Joseph</name>
  <birth>1955</birth>
  <mother refid="margaret"/>
  <father refid="luigi"/>
</node>

<node id="margaret">
  <name>Margaret</name>
  <birth>1923</birth>
</node>
<node id="luigi">
  <name>Luigi</name>
  <birth>1920</birth>
</node>
</tree>
```

# Documentos XML

---

## válidos

Respeitam um conjunto de regras:

- DTD
- XML Schema
- Relax NG

---

# DTDs

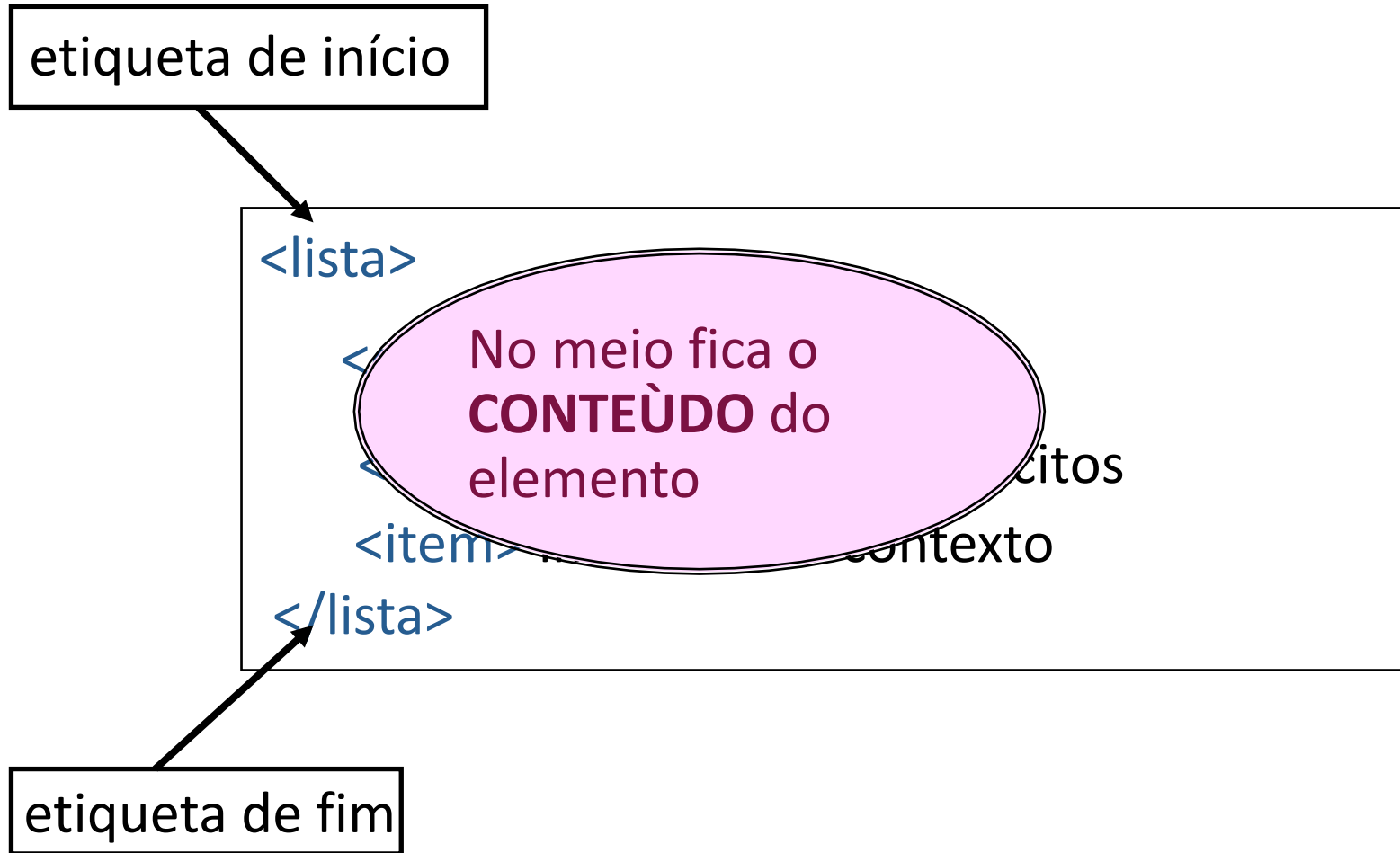
# XML: exemplo

---

```
<lista>  
  <cabec>Os delimitadores podem ser: </cabec>  
  <item> explícitos </item>  
  <item> inferidos do contexto </item>  
</lista>
```

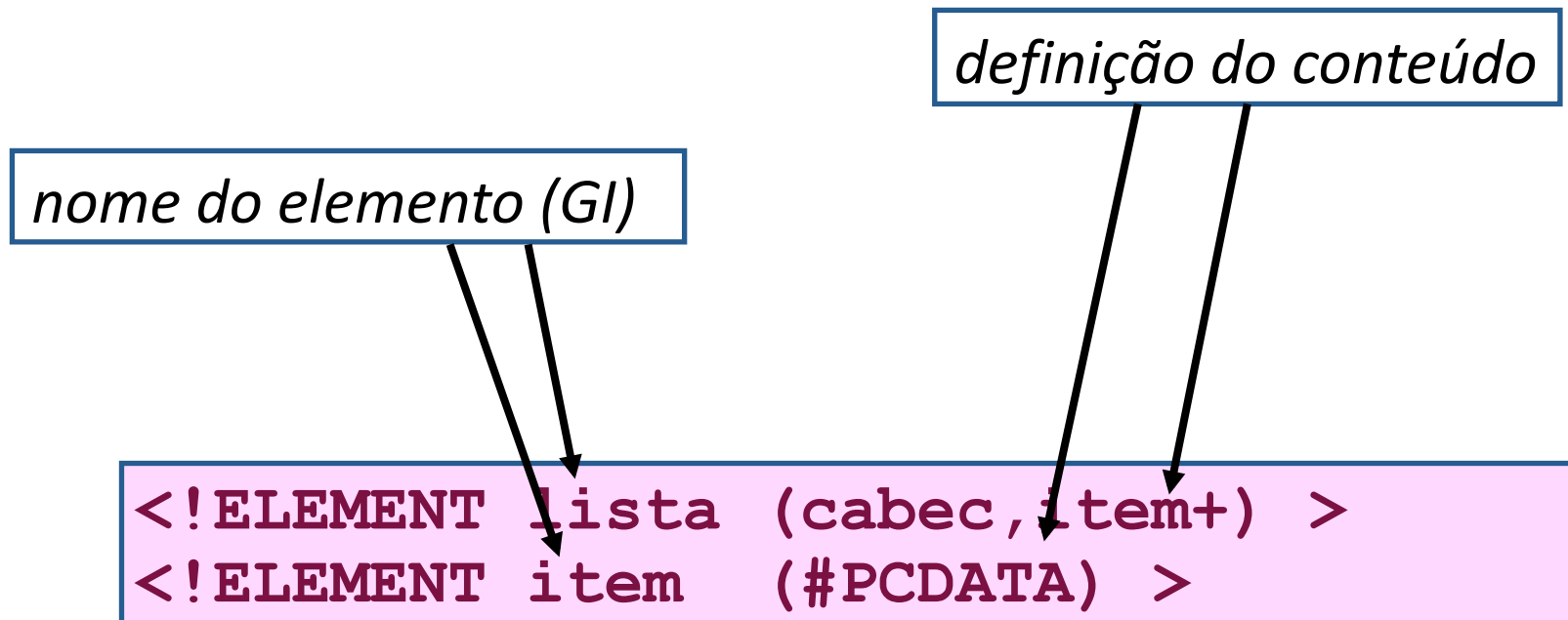
*O elemento do tipo **LISTA** é formado por um elemento **CABEC**, seguido por dois elementos do tipo **ITEM**.*

# Ocorrência de um elemento



# Definição de um elemento

---





# Definição do CONTEÚDO

---

- outros elementos especificados
- ANY (qq elemento especificado)
- EMPTY (nada, vazio)
- #PCDATA (texto)
- uma mistura de elementos com #PCDATA

# Expressão de Conteúdo: sintaxe

---

- sequência
  - $a,b$     *a seguido de b*
  - $a|b$     *a ou b mas não ambos*
- ocorrência
  - $a$             *um e apenas um*
  - $a?$             *opcionalmente um (0 ou 1)*
  - $a^*$             *zero ou mais*
  - $a^+$             *um ou mais*

# Exemplo: o poema

Poema: título, autor, corpo, data

Corpo: quadra, quadra, terno,  
terno.

Quadra: verso, verso, verso,  
verso

Terno: verso, verso, verso

Verso: (texto | nome)+

Nome: texto

```
<!ELEMENT poema (titulo,autor,corpo,data) >  
<!ELEMENT corpo (quadra,quadra,terno,terno) >  
<!ELEMENT quadra (verso,verso,verso,verso) >  
<!ELEMENT terno (verso,verso,verso) >  
<!ELEMENT verso (#PCDATA |nome)* >
```

# Exercício (pausa)

---

**Especificar o DTD para a Agenda**

---

# XML Schema

---

# Motivação

---

- Sintaxe XML
- Suporte para NameSpaces
- data types, abstract data types, references.
- reuse, extension, restriction, ...

# Exemplo: O Poema

---

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:complexType name="Tquadra">
    <xs:sequence>
      <xs:complexType name="Tverso" mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="nome" type="xs:string"/>
          <xs:element name="lugar" type="xs:string"/>
        </xs:choice>
      </xs:complexType>
      <xs:sequence>
        <xs:element name="verso" type="Tverso"/>
        <xs:element name="verso" type="Tverso"/>
        <xs:element name="verso" type="Tverso"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
... </xs:schema>
```

# Tipos Primitivos

---

- String
- Float
- Decimal
- Date
- ...

```
<xs:element name="nome" type="xs:string"/>  
<xs:element name="lugar" type="xs:string"/>
```



# Tipos Derivados

---

- Derivam dos tipos primitivos
- integer (decimal), long (integer), ...
- Por restrição
- Por extensão

# Simple and Complex Types

---

- Os tipos de dados em XML Schema ou são “simple” ou “complex”.
- Um tipo “simple” é um dos tipos básicos: string, date, float, double, timeDuration, ...
- Um tipo “simple” restringe o texto que pode aparecer no valor dum atributo ou no conteúdo dum elemento textual.
- Um elemento que tenha atributos ou elementos filho é do tipo “complex”.
- Um tipo “complex” restringe o conteúdo dum elemento relativamente aos atributos e elementos filho que pode ter.

# Exemplo: complex Type

---

```
<xs:complexType name="Tquadra">
  <xs:sequence>
    <xs:element name="verso" type="Tverso"/>
    <xs:element name="verso" type="Tverso"/>
    <xs:element name="verso" type="Tverso"/>
    <xs:element name="verso" type="Tverso"/>
  </xs:sequence>
</xs:complexType>
```

**Tverso** é uma referência a um “complex type” definido algures no schema  
**Tquadra** é um complex type com nome (por oposição a um anónimo)

# Scope dos elemento

---

- Um elemento pode ser declarado como **global** ou como **local**.
- Um elemento global é declarado como sendo filho de <schema>.
- Um elemento local é declarado algures na estrutura do Schema.
- Os elementos globais podem ser reutilizados por referência.

# Elementos globais

```
<xs:element name="poema">  
  <xs:complexType>
```

```
<xs:element name="quadra">  
  <xs:complexType>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<quadra>
```

```
  <verso>Olha,<nome>Daisy</nome>: quando eu morrer tu hás-de</verso>
```

```
  <verso>dizer aos meus amigos aí de <nome>Londres</nome>,</verso>
```

```
  <verso>embora não o sintas, que tu escondes</verso>
```

```
  <verso>a grande dor da minha morte. Irás de</verso>
```

```
</quadra>
```

Instância Válida

```
</xs:element>
```

```
<xs:element ref = "Terno"/>
```

```
<xs:element ref = "Terno"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

# Cardinalidade

---

```
<xs:complexType name="Tquadra">
  <xs:sequence>
    <xs:element name="verso" type="Tverso"
      minOccurs="4" maxOccurs="4"/>
  </xs:sequence>
</xs:complexType>
```

A cardinalidade pode ser restringida recorrendo aos atributos: **minOccurs** e **maxOccurs**

# Valores para elementos

---

```
<xs:element name="email" type="xs:string"  
            default="jcr@di.uminho.pt"/>
```

Valor por omissão

```
<xs:element name="supervisor" type="xs:string"  
            fixed="José Carlos Ramalho"/>
```

Valor fixo

# Atributos

---

- Os atributos declaram-se recorrendo ao elemento “<attribute>”
- Um atributo é sempre do tipo “simple”

```
<xs:attribute name="tipo"  
              type="xs:string"  
              use="optional"  
              default="soneto"/>
```

use: (optional | default | required | fixed )



# Restrição de tipos primitivos

---

```
<xs:simpleType name="dia" base = "xs:integer">  
  <xs:minInclusive value="1"/>  
  <xs:maxInclusive value="31"/>  
</xs:simpleType>
```

Restrição de domínio

```
<xs:simpleType name="telefone" base = "xs:integer">  
  <xs:pattern value="253-\d{6}"/>  
</xs:simpleType>
```

Restrição de formato

# Restrições ou facetas

---

- minInclusive
- maxInclusive
- maxExclusive
- Length
- minLength
- maxLength
- Pattern
- Enumeration
- minExclusive
- Precision
- Scale
- Encoding
- Period
- Duration

# Restrição por enumeração

---

```
<xs:simpleType name="diaSemana" base = "xs:string">  
  <xs:enumeration value="Domingo"/>  
  <xs:enumeration value="Segunda"/>  
  <xs:enumeration value="Terça"/>  
  <xs:enumeration value="Quarta"/>  
  <xs:enumeration value="Quinta"/>  
  <xs:enumeration value="Sexta"/>  
  <xs:enumeration value="Sábado"/>  
</xs:simpleType>
```

# Elementos vazios

---

```
<xs:element name="ref">
  <xs:complexType>
    <xs:element name="imagem">
      <xs:complexType>
        <xs:attribute name="path" type="xs:string"
          use="required"/>
        <xs:attribute name="formato" type="xs:string"
          use="optional"/>
      </xs:complexType>
    </xs:element>
  </xs:complexType>
</xs:element>
```

# Sessão Prática

---

1. Definição de Linguagens de Anotação
  - a) Uma agenda de contactos
  - b) Um anúncio de evento
2. Criação de Instâncias documentais
3. Validação
4. Query: Xpath
5. Trabalho de Grupo
  - Definição de uma linguagem de anotação para agendas de eventos

# Trabalho de Grupo

---

Pretende-se desenvolver um sistema para gestão de eventos. A informação relativa aos vários eventos é armazenada num documento XML.

Desenvolva então as seguintes alíneas:

(a) Especifique um Schema para este tipo de documentos XML atendendo aos seguintes pressupostos:

- \* Num documento podem estar descritos vários eventos.
- \* Um evento tem uma designação/título, uma descrição, o tipo (desportivo, cultural, ...), a data e hora em que foi registado, o período (data de início e data de fim) em que deverá "estar vivo", e a informação da pessoa que o registou.
- \* Uma descrição é um texto que pode conter imagens e referências a outros eventos.

(b) Defina uma instância documental com a linguagem de anotação que acabou de definir.

---

# Web Semântica

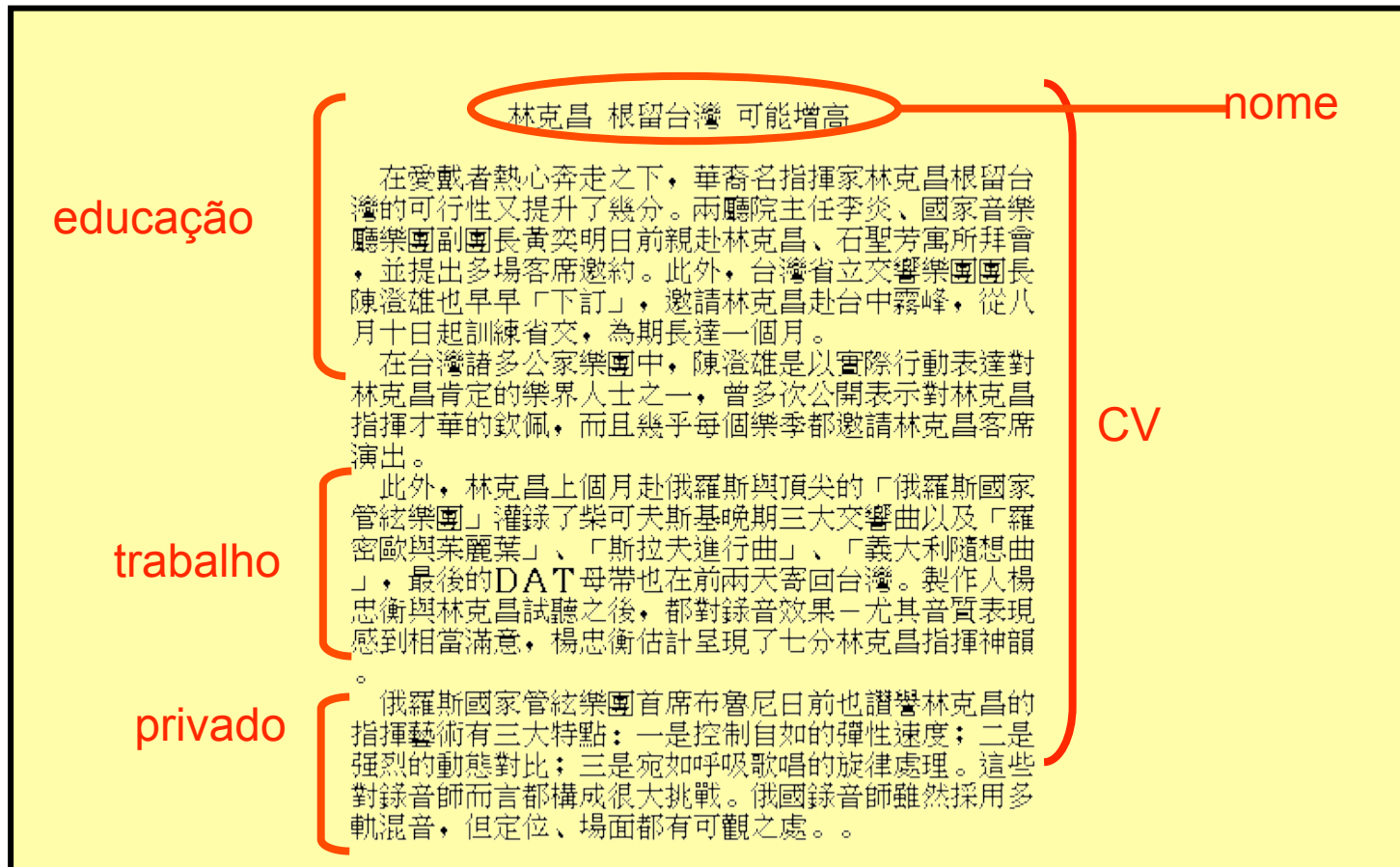
---

„The **Semantic Web** is an extension of the current web in which information is given well-defined **meaning**, better enabling computers and people to **work in co-operation**.“

[Berners-Lee et al., 2001]



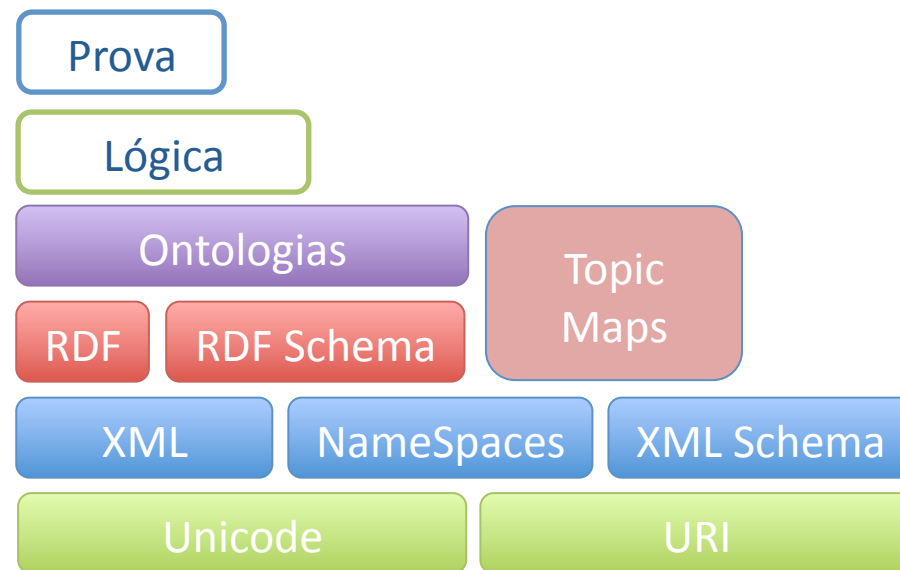
# O que a máquina vê





# Camadas da Web Semântica

---



# XML

---

```
<H1>Knowledge Management</H1>  
<UL>  
  <LI>Teacher: Rudi Studer  
  <LI>Students: Master  
</UL>
```

HTML:

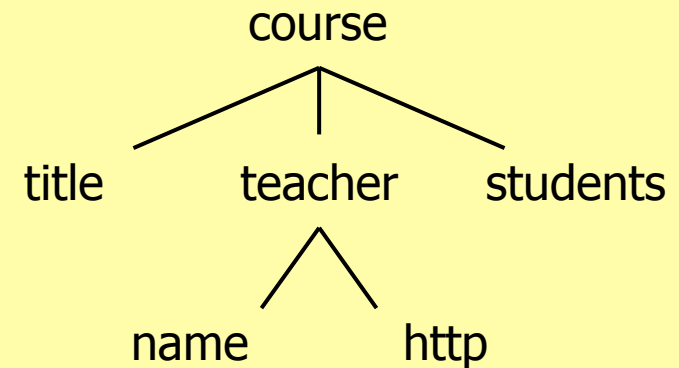
```
<course>  
  <title>Knowledge Management</title>  
  <teacher>Rudi Studer</teacher>  
  <students>Master</students>  
</course>
```

XML:

# XML: Document = labelled tree

```
<course date="...">  
  <title>...</title>  
  <teacher>...</teacher>  
    <name>...</name>  
    <http>...</http>  
  <students>...</students>  
</course>
```

=



- **DTD**: gramáticas simples
- Então:

**Porque não usar XML para as Ontologias?**

# Limitações do XML

---

O XML não se compromete com:

- ① Um **vocabulário** específico para ontologias
- ② **Primitivas para modelar ontologias**

⇒ requer um pré-acordo sobre ① & ②

Apenas viável em colaborações fechadas

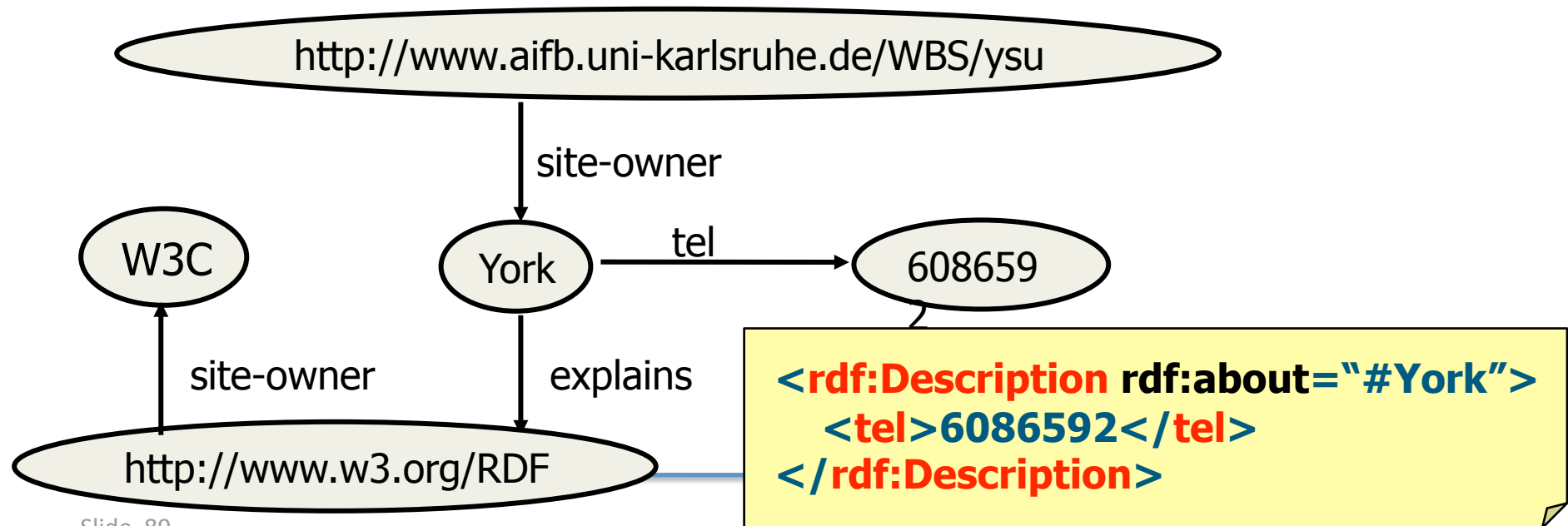
- agentes numa comunidade pequena e estável
- páginas numa intranet estável

**...não para recursos Web que se querem  
partilhar**



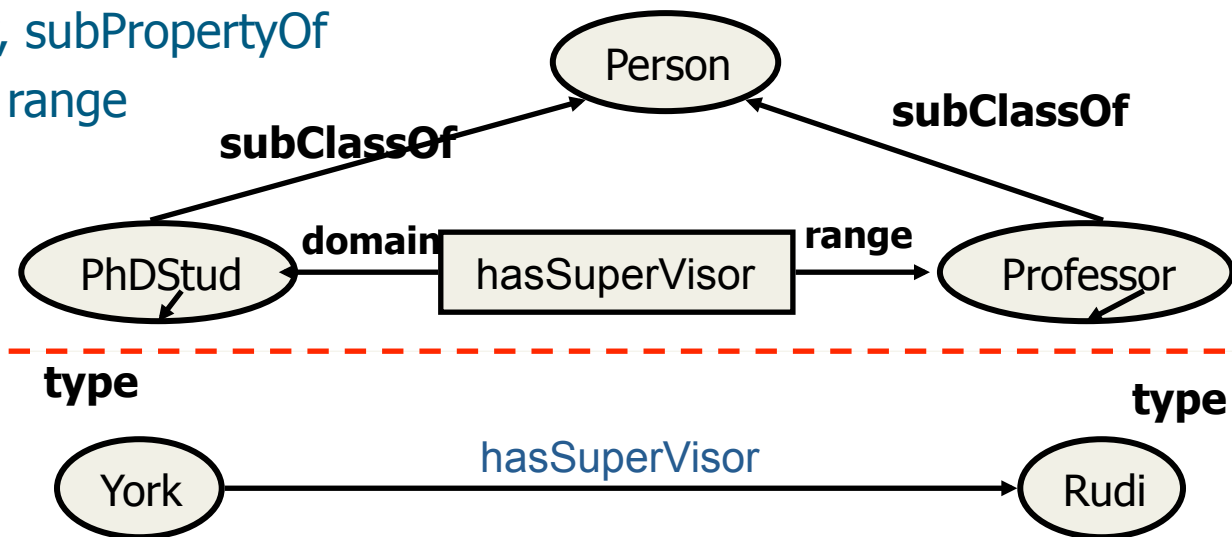
# RDF

- RDF provides metadata about Web resources
- **Object -> Attribute-> Value** triples
- It has an **XML syntax**
- Chained triples form a **graph**



# What does RDF Schema add?

- Defines **vocabulary** for RDF
- Organizes this vocabulary in a **typed hierarchy**
  - Class, subClassOf, type
  - Property, subPropertyOf
  - domain, range



# RDF Schema

```
<rdf:Description ID="MotorVehicle">  
  <rdf:type resource="http://www.w3.org/...#Class"/>  
  <rdfs:subClassOf rdf:resource="http://www.w3.org/...#Resource"/>  
</rdf:Description>
```

```
<rdf:Description ID="Truck">  
  <rdf:type resource="http://www.w3.org/...#Class"/>  
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>  
</rdf:Description>
```

```
<rdf:Description ID="registeredTo">  
  <rdf:type resource="http://www.w3.org/...#Property"/>  
  <rdfs:domain rdf:resource="#MotorVehicle"/>  
  <rdfs:range rdf:resource="#Person"/>  
</rdf:Description>
```

```
<rdf:Description ID="ownedBy">  
  <rdf:type resource="http://www.w3.org/...#Property"/>  
  <rdfs:subPropertyOf rdf:resource="#registeredTo"/>  
</rdf:Description>
```

# Conclusions about RDF(S)

---

- Next step up from plain XML:
  - (small) **ontological commitment** to modeling primitives
  - possible to define **vocabulary**
- However:
  - **no** precisely described meaning
  - **no** inference model



# Ontology

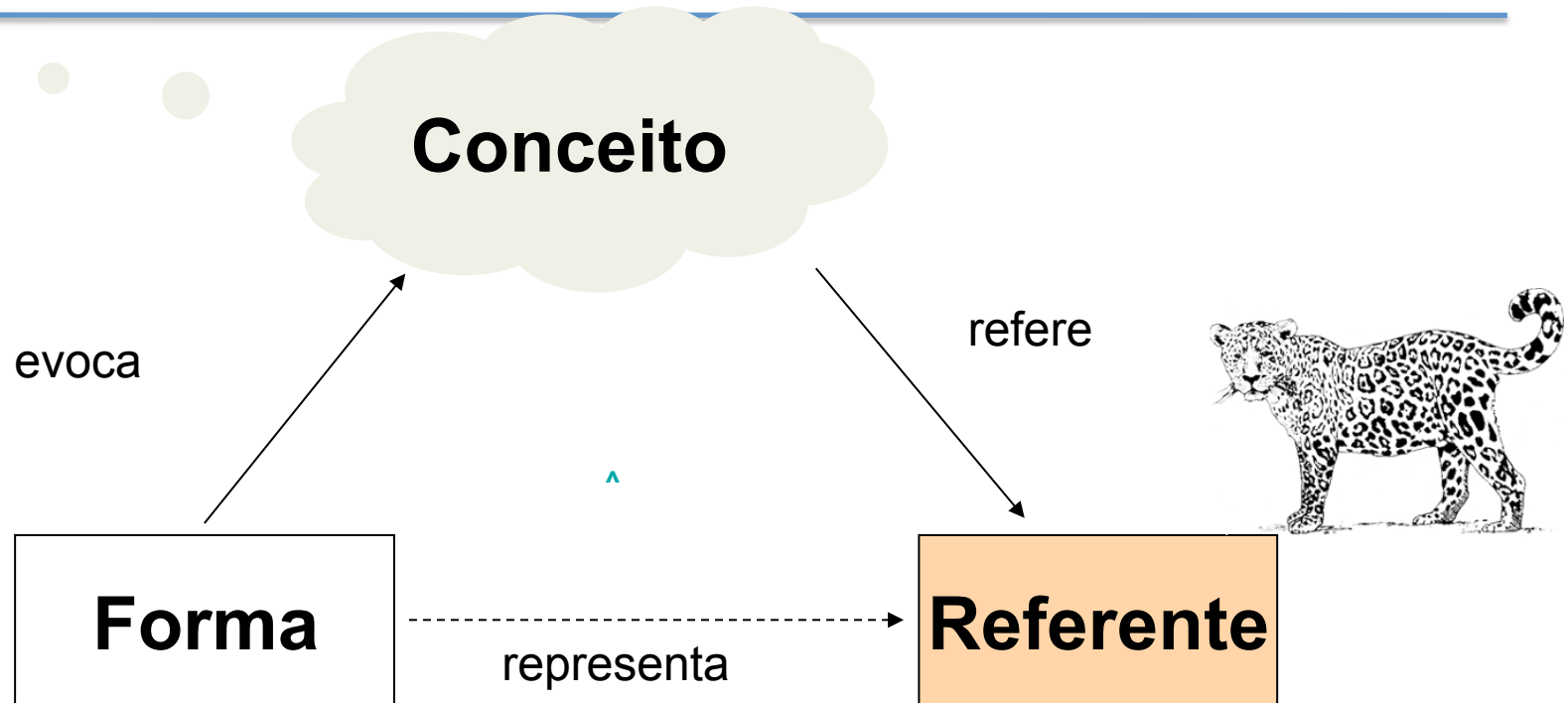
---

„People can't **share knowledge** if they do not speak a **common language.**“ [Davenport & Prusak, 1998]

„An ontology is an **explicit specification** of a **conceptualization.**“ [Gruber, 1993]

- Ontologies enable a **better communication** between Humans/Machines
- Ontologies **standardize** and **formalize** the meaning of words through concepts

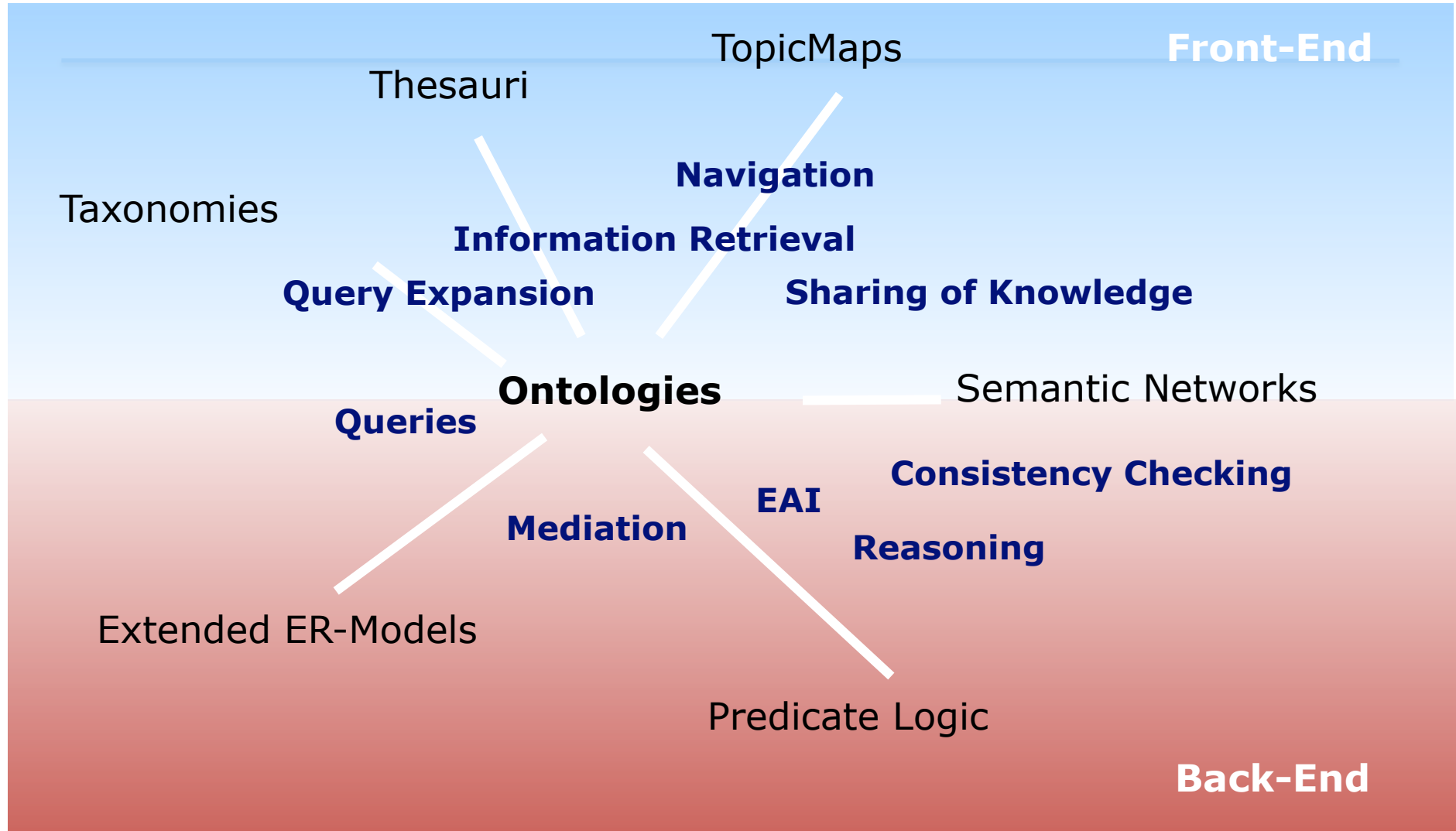
# Triângulo do Significado



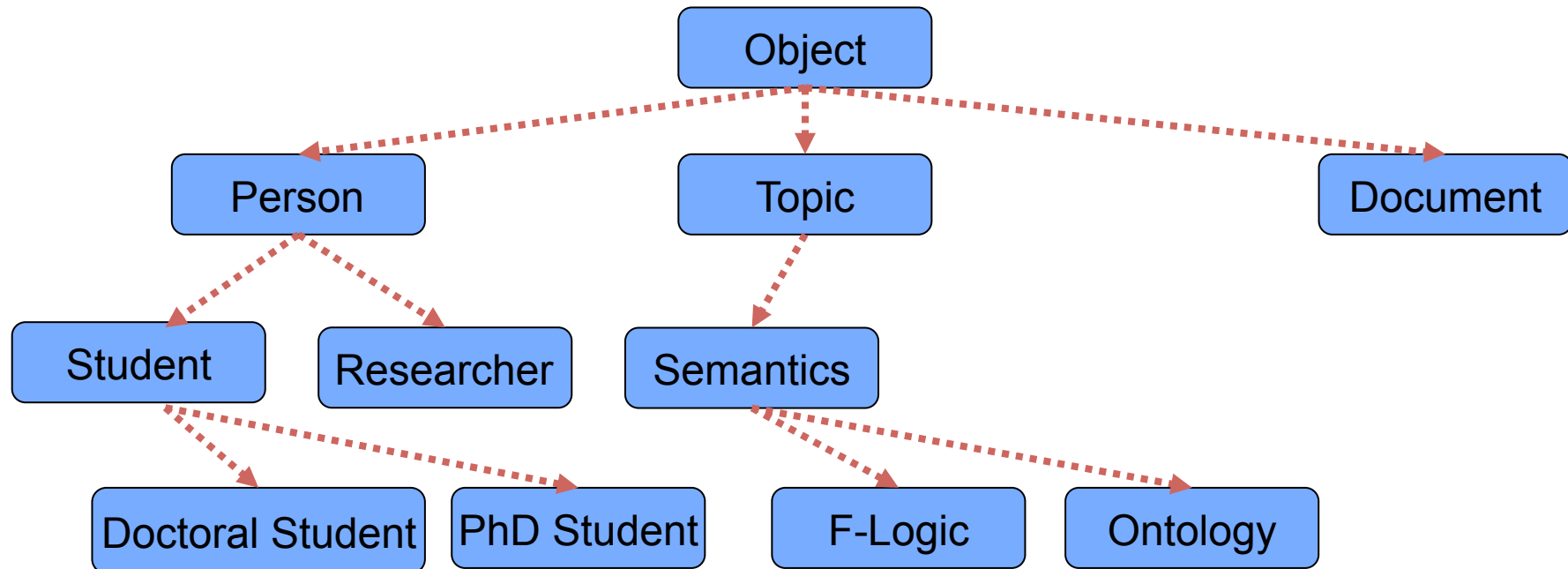
***“Jaguar”***

**[Odwen, Richards, 1923]**

# Views on Ontologies

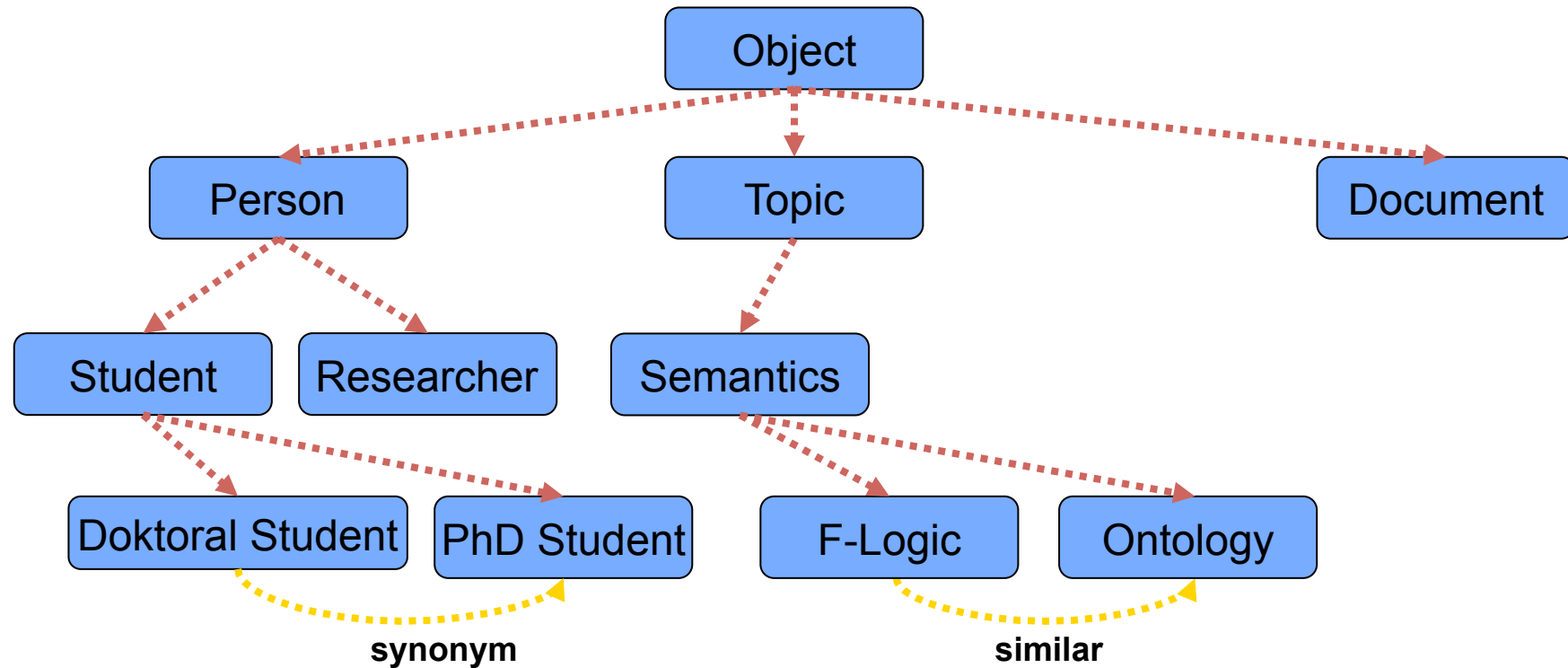


# Taxonomy



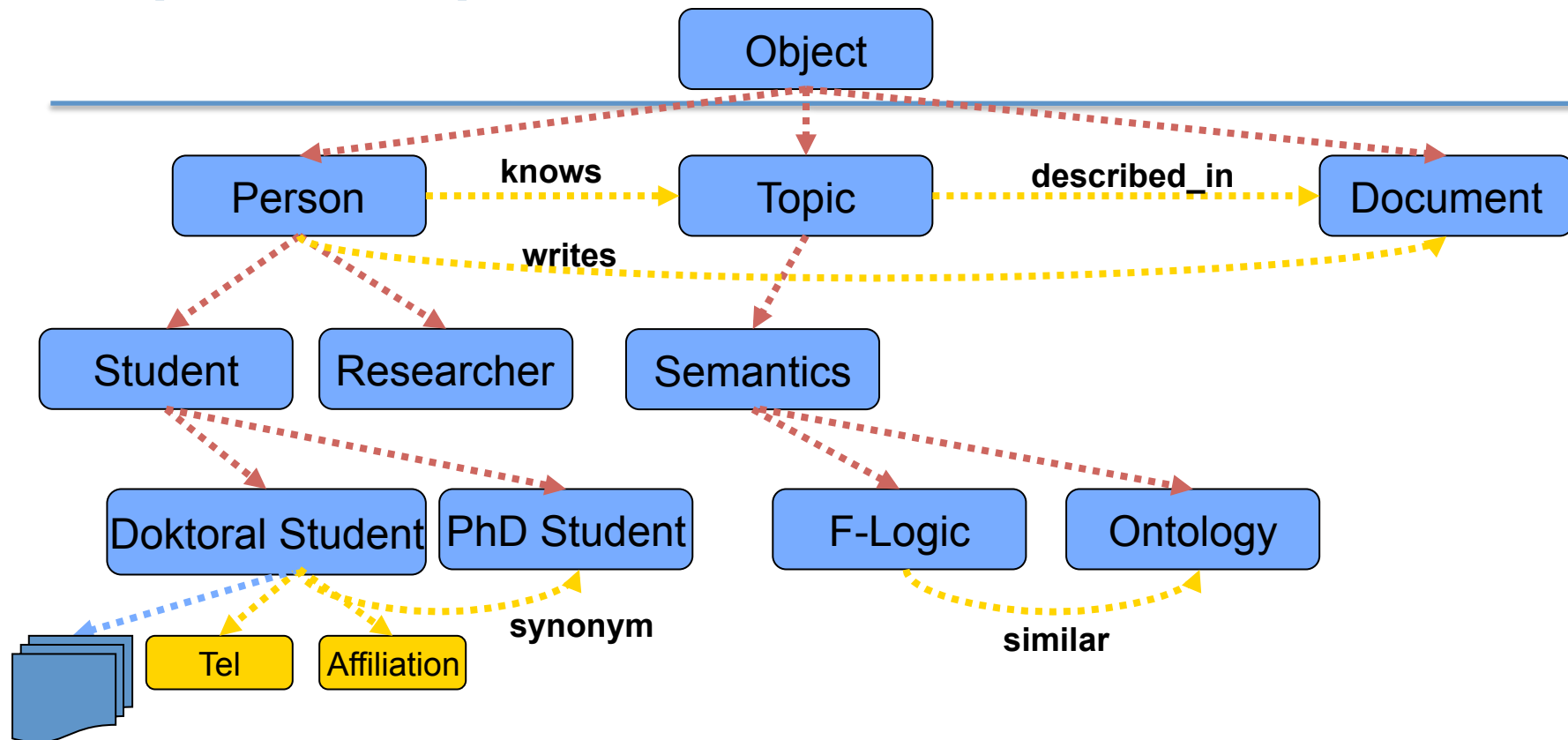
**Taxonomy := Segementation, classification and ordering of elements into a classification system according to their relationships between each other**

# Thesaurus



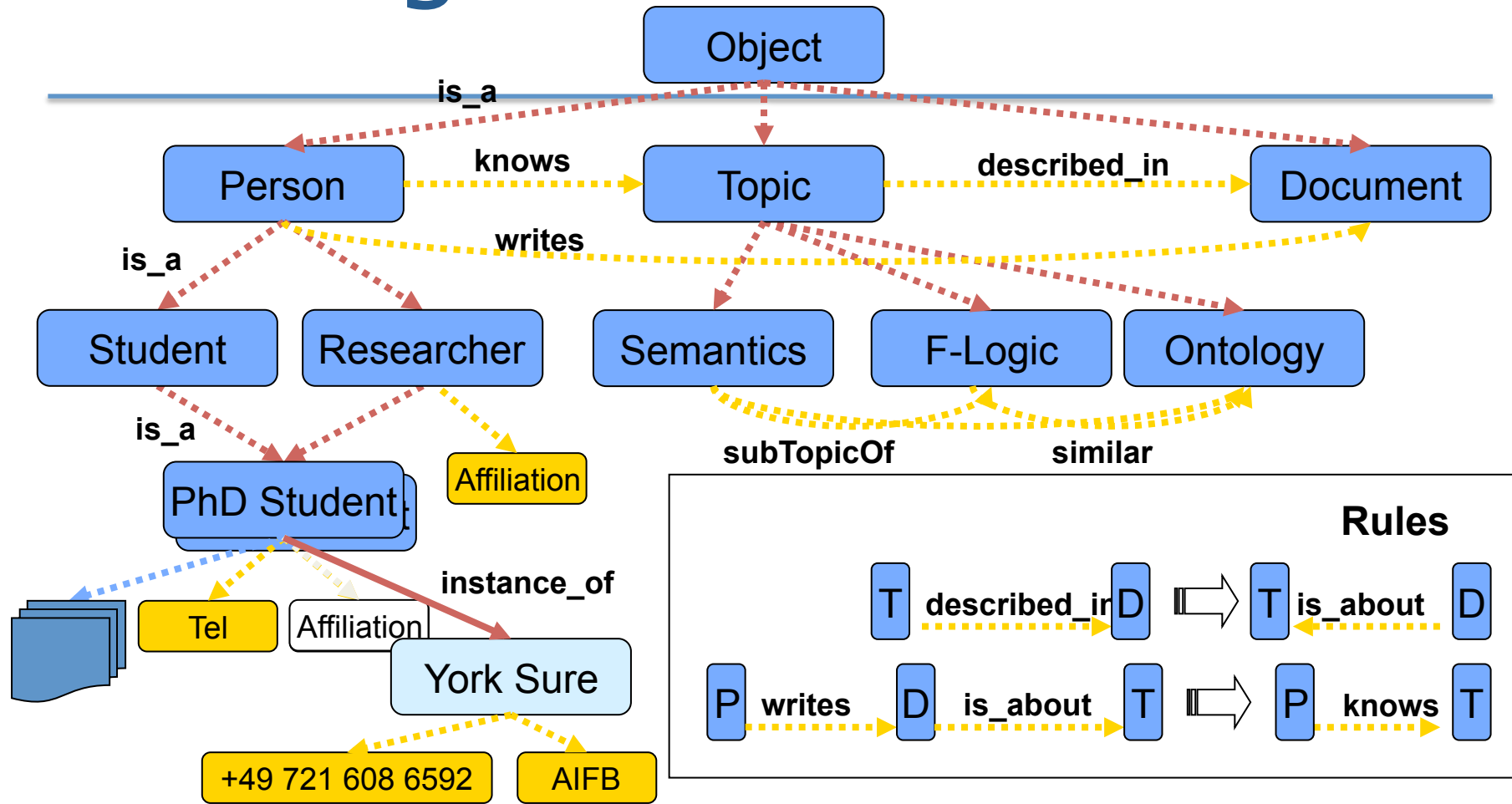
- Terminology for specific domain
- Graph with primitives, 2 fixed relationships (similar, synonym)
- originate from bibliography

# Topic Map



- Topics (nodes), relationships and occurrences (to documents)
- ISO-Standard
- typically for navigation- and visualisation

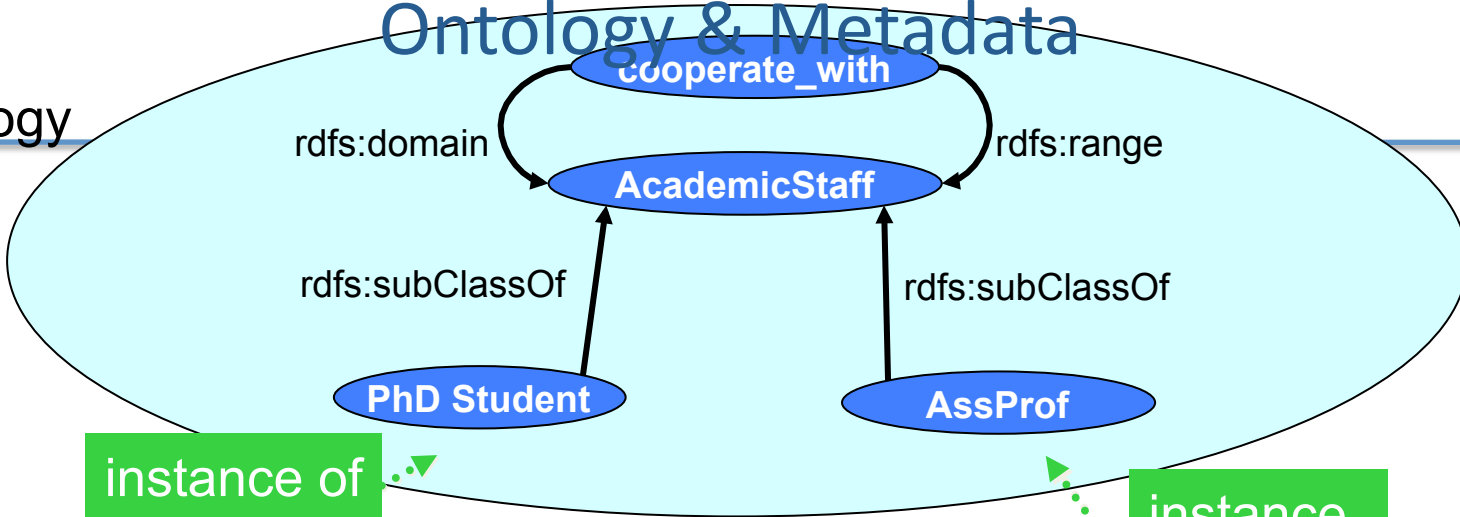
# Ontologia



- Representation Language: Predicate Logic (F-Logic)
- Standards: RDF(S); OWL

# Ontology & Metadata

Ontology



instance of

instance of

Annotation

```

<swrc:PhD_Student rdf:ID="sha">
  <swrc:name>Siegfried Handschuh</swrc:name>
  <swrc:cooperate_with rdf:resource =
    "http://www.aifb.uni-karlsruhe.de/WBS/sst#sst"/>
  ...
  
```

```

<swrc:AssProf rdf:ID="sst">
  <swrc:name>Steffen Staab</swrc:name>
  ...
</swrc:AssProf>
  
```

Cooperate\_with

Web Page

**Siegfried Handschuh**



He is working together with Steffen Staab in the Knowledge Management Group

Links have explicit meanings!



**Research:**

Semantic Web, Knowledge Management, Natural Language,

URL

<http://www.aifb.uni-karlsruhe.de/WBS/sha>

<http://www.aifb.uni-karlsruhe.de/WBS/sst>



# Para que se definem Ontologias?

---

- Para partilhar o conhecimento sobre a estrutura da informação entre pessoas ou agentes de software;
- Para tornar o conhecimento reutilizável;
- Para tornar as asserções explícitas
- Para analisar o domínio do conhecimento

# Definição de Ontologias

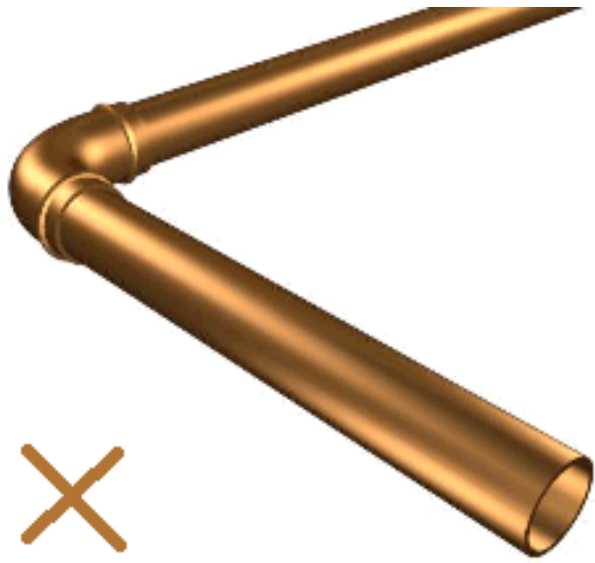
---

1. Definição do domínio
2. Enumeração dos termos
3. Definição da hierarquia de classes
4. Definição das propriedades de cada classe
5. Criação de instâncias

---

# Ontologias

(em síntese)



---

this is not  
a pipe



---

Google™

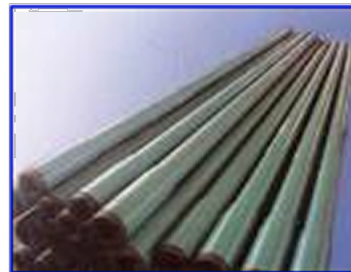
pipe

Search Images



*Ceci n'est pas une pipe.*

**This is Not a Pipe (1968)**  
510 x 364 - 25k - jpg  
[foucault.info](http://foucault.info)



**Drill Pipe**  
500 x 379 - 35k - jpg  
[www.made-in-china.com](http://www.made-in-china.com)



**Hello Kitty Exhaust Pipe.**  
432 x 322 - 408k - bmp  
[www.pogadget.net](http://www.pogadget.net)

---

Google™

orange

Procurar imagens

[SafeSearch Moderado está activo](#)



Guilherme I, príncipe de **Orange**  
300 x 383 - 14k - jpg  
[pt.pandapedia.com](http://pt.pandapedia.com)



.. celular europeia **Orange** estariam ...  
450 x 487 - 43k - jpg  
[vidaportatil.com.br](http://vidaportatil.com.br)



Banco de fotografias: **Orange**  
380 x 380 - 56k - jpg  
[portuguesbrasileiro.istockphoto.com](http://portuguesbrasileiro.istockphoto.com)  
[ [Mais a partir de](#)  
[www1.istockphoto.com](http://www1.istockphoto.com) ]

---

Nós interpretamos



as máquinas não!

Mozilla

Fichier Edition Affichage Aller à Marque-pages Outils Fenêtre Aide

http://www.reazyread.com/sacks.html



## The Man Who Mistook His Wife for a Hat : And Other Clinical Tales by [Oliver W. Sacks](#)

In his most extraordinary book, "one of the great clinical writers of the 20th century" ([The New York Times](#)) recounts the case histories of patients lost in the bizarre, apparently inescapable world of neurological disorders. Oliver Sacks's *The Man Who Mistook His Wife for a Hat* tells the stories of individuals afflicted with disorders of memory, perception, and identity. Some have memories that are too good to be true; others have lost their memories and cannot learn from their experiences. Some have strange perceptions of the world around them; others have lost their sense of self. Some are gifted with unusual talents; others are afflicted with unusual disabilities. Some are brilliant; others are ordinary. Some are happy; others are sad. Some are loved; others are not. Some are famous; others are not. Some are heroes; others are not. Some are... [The New York Times](#)

If inconceivably strange, these brilliant tales are studies of life struggling against increasingly impaired, to imagine with our hearts what medicine's ultimate responsibility: "the su

Our rating : ★★★★★

Find other books in :  Neurology

Search books by terms :

Google - Mozilla

Fichier Edition Affichage Aller à Marque-pages Outils Fenêtre Aide

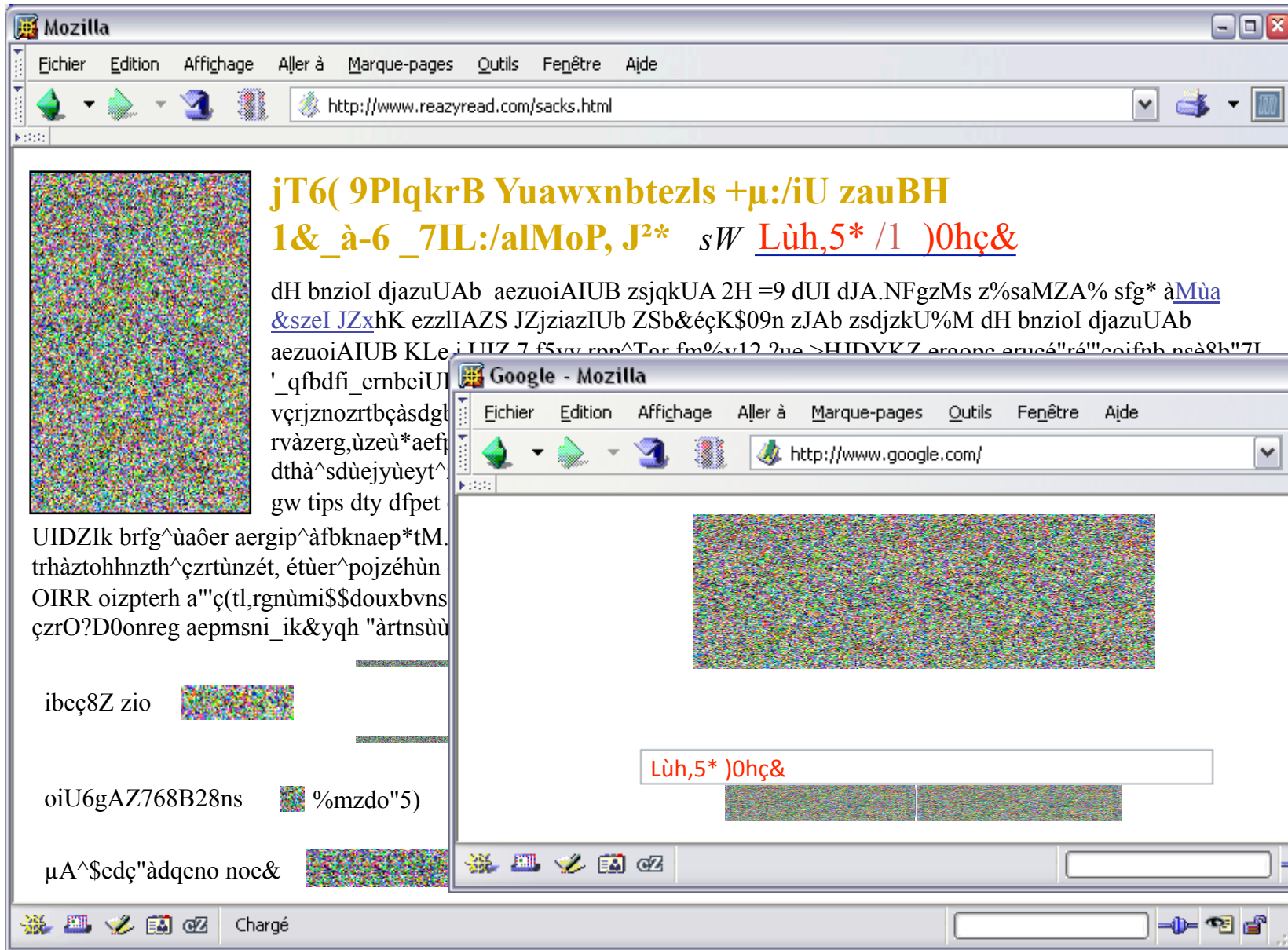
http://www.google.com/



Google Search I'm Feeling Lucky

Chargé

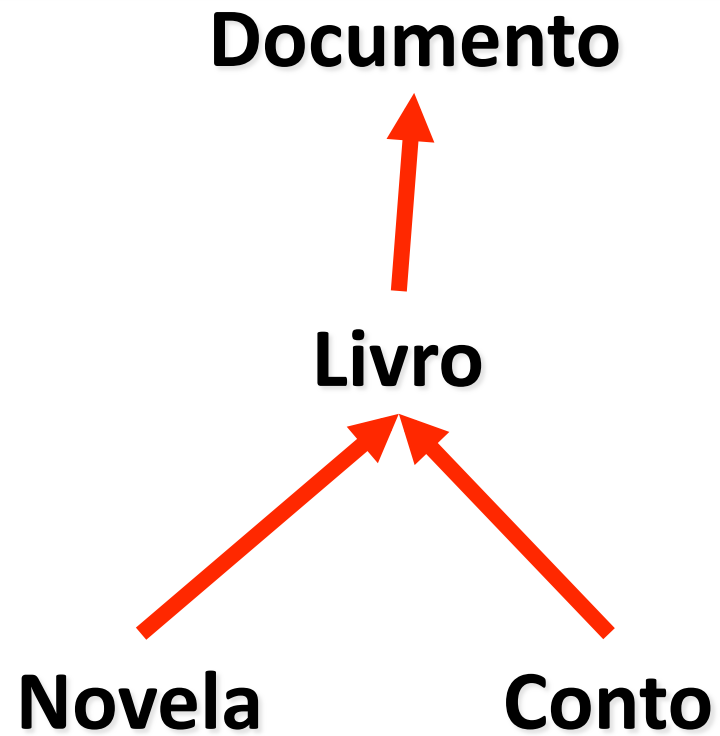




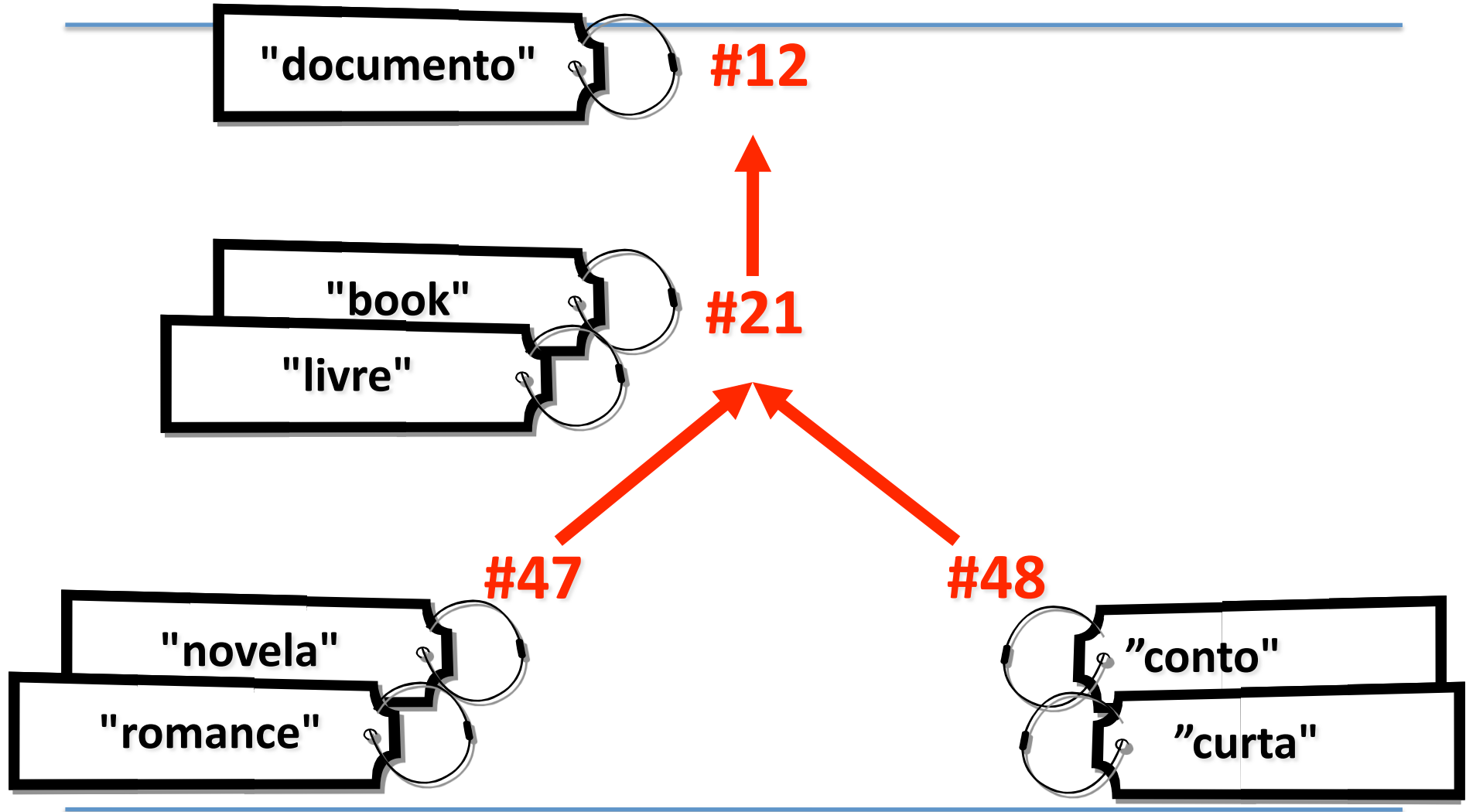
---

falta alguma coisa  
algo designado por “conhecimento”

is  
a

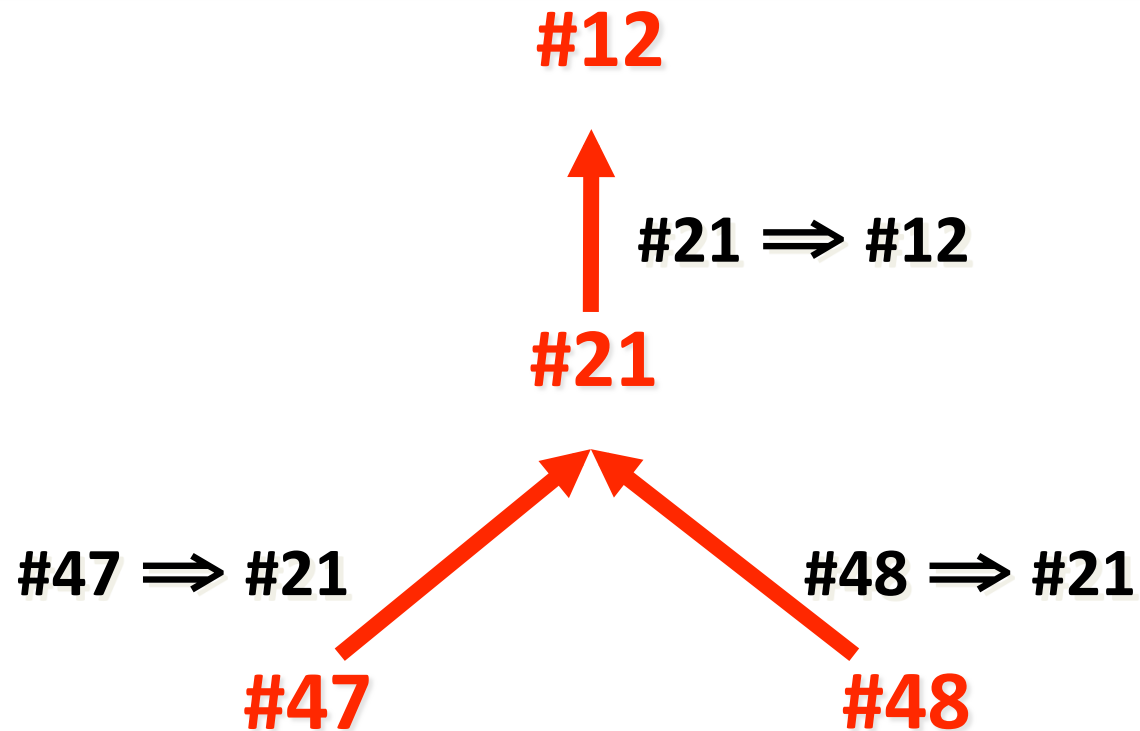


is  
a



# Conhecimento ontológico formalizado

---



# Especificação de significado com identificadores únicos

---



---

# Ontologia

não é um sinónimo de

# **Taxonomia**

---

# As Taxonomias

são uma espécie de

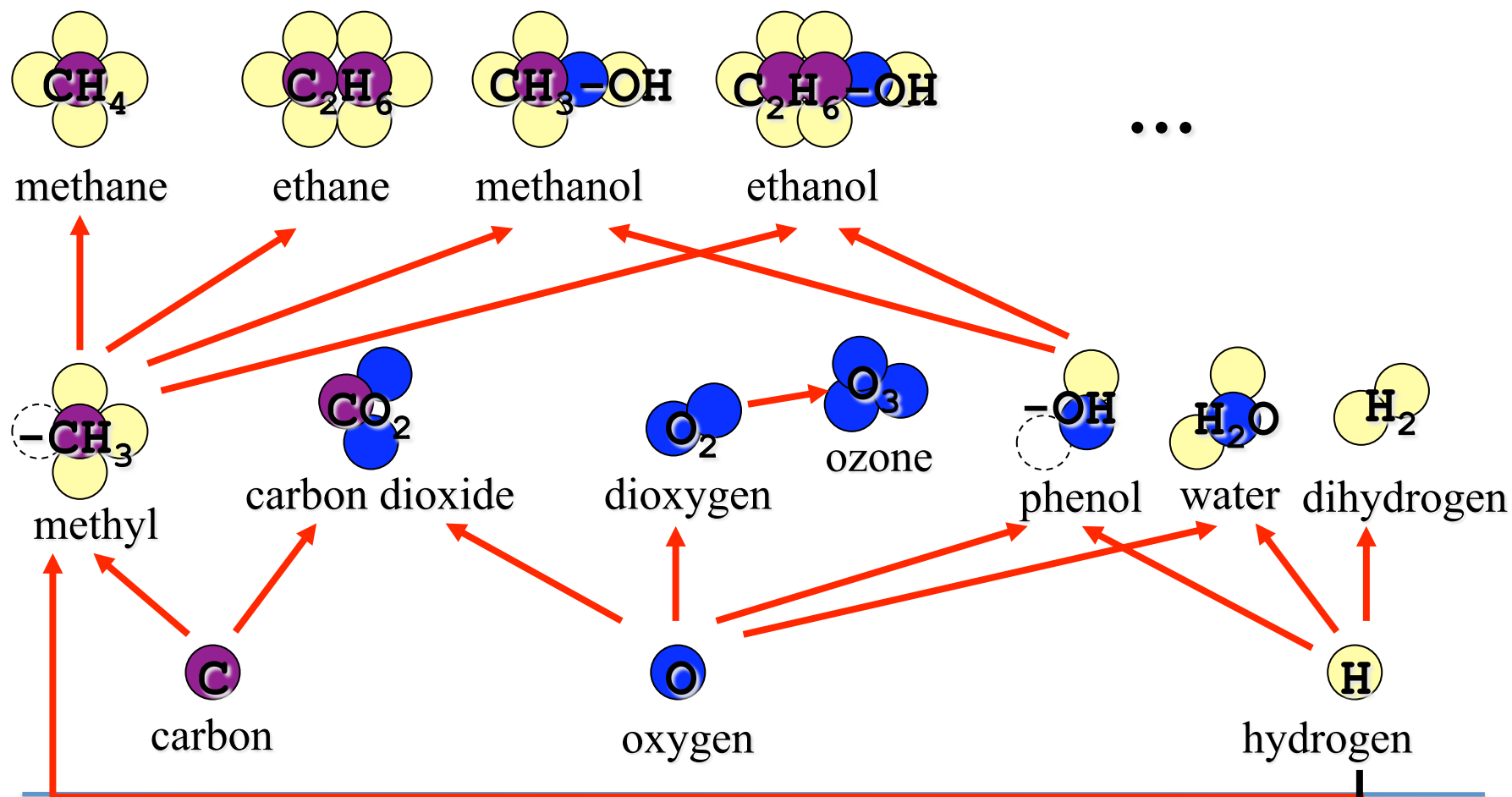
# Ontologias

entre outras

---



part  
of





---

# O ntologia

- uma teoria lógica que dá uma interpretação explícita e parcial de uma conceptualização, *i.e.* uma estrutura semântica que contem as regras implícitas que restringem uma parte da realidade;
- o objectivo de uma ontologia é definir as primitivas, que com a respectiva semântica associada, são necessárias à representação do conhecimento de um determinado contexto.  
[Gruber, 1993] [Guarino & Giaretta, 1995] [Bachimont, 2000]

# Domínio/cobertura

---



extensão coberta pela ontologia.

# Especificidade

---

até que ponto as primitivas  
ontológicas conseguem  
dar uma identificação  
precisa.



# Granularidade

---

até que ponto as primitivas têm uma  
definição precisa e formal.



# Formalização

---

até que ponto as primitivas estão descritas numa linguagem formal.



# Conhecimento (implícito?)

---

- Em qualquer área do conhecimento, a compreensão mútua baseia-se numa **conceptualização partilhada** desenvolvida por académicos;
  - Esta conceptualização partilhada está muitas vezes implícita na escolha do vocabulário e das teorias usadas pelos académicos no seu discurso e na sua escrita;
  - Para comunicar ao mais alto nível (por exemplo num artigo científico), os autores assumem que quem os ouve ou lê pertencem à sua comunidade e partilham a mesma conceptualização da realidade;
  - A maior parte da mensagem numa comunicação científica é em primeiro lugar um reforço do conhecimento implícito e em segundo uma extensão do conhecimento partilhado;
  - É, às vezes, este conhecimento implícito que torna a literatura científica impenetrável àqueles que não são especialistas.
-

# Comunicação electrónica de conhecimento

---

- Na sociedade humana, *muito do nosso conhecimento é implícito* - sabemos mais do que pensamos saber!
  - No entanto, hoje, com cada vez mais informação e conhecimento a serem colocados em linha, mais a comunicação tem de ser feita entre máquinas M2M;
  - Para se conseguir isto será necessário:
    - que o conhecimento implícito se torne explícito, e
    - que o significado da informação seja especificado sem ambiguidades;
-



---

# Alguns exemplos



---

```
(define-class human (?human)
  :def (animal ?human) )
```

example

subsumption in frames

---

---

```
<Class rdf:ID="Man">
  <subClassOf rdf:resource="#Person"/>
  <subClassOf rdf:resource="#Male"/>
  <label xml:lang="en">man</label>
  <comment xml:lang="en">an adult male
                                person</comment>
</Class>
```

example

a class declaration in RDFS

---

---

```
(defprimconcept MALE)
(defprimconcept FEMALE)
(disjoint MALE FEMALE)
```

example

disjoint classes in description logics

---

---

```
<owl:Class rdd:id="AuthorAgent">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Class rdf:about="#Group"/>
  </owl:unionOf>
</owl:Class>
```

example

union of classes in OWL

---

---

```
<owl:Class rdf:ID="Man">  
  <owl:intersectionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#Male"/>  
    <owl:Class rdf:about="#Person"/>  
  </owl:intersectionOf>  
</owl:Class>
```

example

intersection of classes in OWL

---

---

```
<owl:Class rdf:id="EyeColor">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:ID="Blue"/>
    <owl:Thing rdf:ID="Green"/>
    <owl:Thing rdf:ID="Brown"/>
  </owl:oneOf>
</owl:Class>
```

example

enumerated class in OWL

---

---

```
<owl:Class rdf:ID="Male">  
  <owl:complementOf rdf:resource="#Female"/>  
</owl:Class>
```

example

complement of classes in OWL

---



---

[Concept: **Director**] -> (Def) ->

[LambdaExpression:

[**Person**:  $\lambda$ ] -> (Manage) -> [**Group**]

example

defined class in conceptual graphs

---

---

```
<rdf:Property rdf:ID="hasMother">
  <subPropertyOf rdf:resource="#hasParent" />
  <range rdf:resource="#Female" />
  <domain rdf:resource="#Human" />
  <label xml:lang="en">has for mother</label>
  <comment xml:lang="en">to have for parent a
                          female.</comment>
</rdf:Property>
```

example

declare a property in RDFS

---

---

```
(define-relation has-mother
                 (?child ?mother)
:iff-def
  (and (has-parent ?child ?mother)
        (female ?mother)))
```

example

define a relation in frames

---

---

```
<owl:Class rdf:ID="Herbivore">
  <subClassOf rdf:resource="#Animal"/>
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eats" />
      <owl:allValuesFrom rdf:resource="#Plant" />
    </owl:Restriction>
  </subClassOf>
</owl:Class>
```

example

restriction on properties in OWL

---

---

```
(define-class executive (?person)
  :default-constraints
  (owns-tv ?person))
```

example

default values in ontolingua

---

---

```
(define-class Author (?author)
  :def (and (person ?author)
    (= (value-cardinality ?author
                          author.name) 1)
    (value-type ?author author.name
                biblio-name)
    (>= (value-cardinality ?author
                          author.documents) 1)
    (<=> (author.name ?author ?name)
          (person.name ?author ?name))))
```

example

cardinality constraints in frames

---

---

```
<owl:SymmetricProperty rdf:ID="hasSpouse" />
<owl:TransitiveProperty rdf:ID="hasAncestor" />
<owl:FunctionalProperty rdf:ID="hasMother" />
<owl:InverseFunctionalProperty rdf:ID="SSNum" />

<rdf:Property rdf:ID="hasChild">
  <owl:inverseOf rdf:resource="#hasParent"/>
</rdf:Property>
```

example

algebraic properties in OWL

---

---

[Car: **∃**] -> (Has) -> [**SteeringWheel**]

example

existential knowledge in conceptual graphs

---



---

```
(define-axiom driver-consistency :=  
  (<=> (drive ?a ?p) (driver ?a ?p))
```

example  
axioms in frames

---

---

```
(defrelation child
  ((?p Person) (?c Person))
  :=> (> (age ?p) (age ?c)))
```

example

constraints in description logics

---

---

```
(define-function price (?car ?power ?days)
                        :-> ?amount
: def (and (Car ?car) (Number ?power)
          (Number ?days) (Number ?amount)
          (Rate ?car ?rate))
: lambda-body
  (* (+ ?rate (* 0.1 ?power)) ?days))
```

example

functions in conceptual graphs

---

---

**IF**

?person author ?doc  
?doc rdf:type PhDThesis  
?doc concern ?topic

**THEN**

?person expertIn ?topic  
?person rdf:type PhD

**example**

derivation rule languages

---

# The BioImage advanced search interface

The screenshot shows the BioImage advanced search interface within a Mozilla Firebird browser window. The browser's address bar displays the URL <http://www.bioimage.org/retrieval/search.do>. The page header includes the BioImage logo and navigation links: ABOUT BIOIMAGE, USER GUIDE, CONTACT US, DOWNLOAD BASKET, and GO TO E-BIOSCI. The main content area is divided into three sections:

- New Search:** A grey box containing instructions: "Fill in at least one field. Fill in more to narrow your search. You do not need to fill in all the fields. Use AND, OR and NOT to combine terms. Search terms are not case sensitive. For further help go to [Search tips](#)".
- Advanced Search:** A blue box with the following fields:
  - Word(s)** (e.g. porin, actin, lymphocyte): Includes radio buttons for "Words anywhere" (selected), "Title", and "Author", and a "Subject" dropdown menu.
  - Bioimage Identifier** (Use the numeric part of the LSID): A text input field.
  - Year(s) of image capture** (e.g. 2001, or 1990-1998): A text input field containing "any".
  - A "Submit" button.
- Your most recent searches:** A blue box with a "Help" link and a list of recent searches:
  - #1 Porin - 12 results [[view](#)]
  - #2 Phi29 - 3 results [[view](#)]
  - #3 Carazco - 8 results [[view](#)]Below the list is a link: "Combine searches or add additional terms to an existing search".

At the bottom of the page, a copyright notice reads: ©2004 BioImage. All rights reserved. Statement of purpose, access and submission policies, disclaimer and copyright details.

The Advanced Search Interface permits Boolean searches, search restrictions, and re-use of previous searches in combination with new terms

# Geração automática de SQL

## Parameters:

```
SELECT Grant.grantNumber
FROM Study
WHERE Experiment.technique=TransmissionElectronMicroscopy
```

Submit Query

## SQL String:

```
SELECT _foo._path, _foo._source, _bar.* FROM (SELECT _left._source AS _source, _right._target AS _target, _left._path||'.study_has_expe
FROM ((SELECT id AS _source, id AS _target, id AS _ref, 'Study-'||id::text AS _path FROM "study") AS _left JOIN (SELECT _left._source AS
_right._target AS _target, _left._path||'.experiment_has_grant.'||_right._path::text AS _path FROM ((SELECT id AS _source, id AS _target,
'Experiment-'||id::text AS _path FROM "experiment" WHERE (technique='TransmissionElectronMicroscopy')) AS _left JOIN (SELECT * FROM
_left._source=_m2m.experiment1 JOIN (SELECT id AS _source, id AS _target, id AS _ref, 'Grant-'||id::text AS _path FROM "grant") AS _rig
_right ON _left._source=_right._ref)) AS _foo JOIN (SELECT "grant".id , "grant".grantNumber AS grant_grantNumber FROM "grant") AS _bar
```

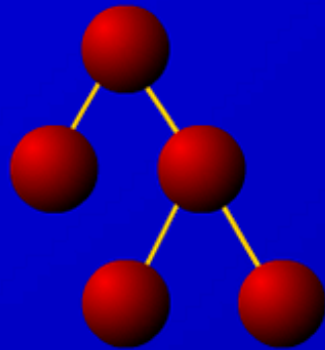
**Time: 0.036 s**

## XML Output:

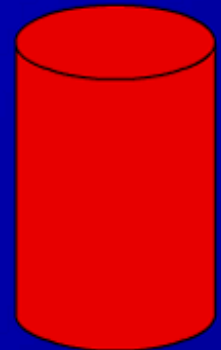
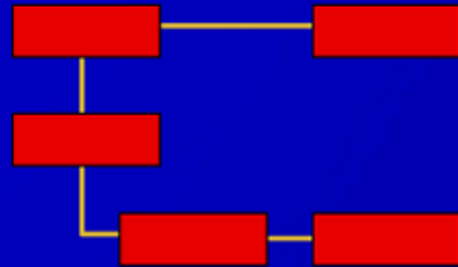
```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <resource id="2" name="Study">
    <relation name="study_has_experiment">
      <resource id="2" name="Experiment">
        <relation name="experiment_has_grant">
          <resource id="2" name="Grant">
```

# (1): query “Encontrar imagens de ursos”

---

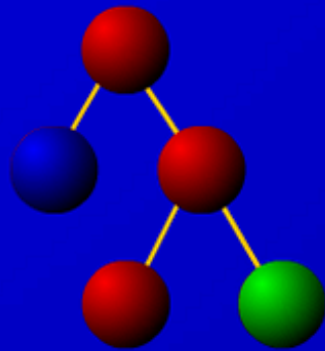


subject = bear

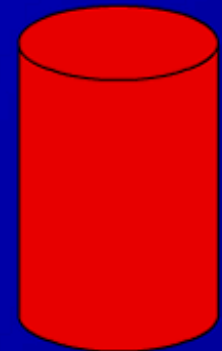
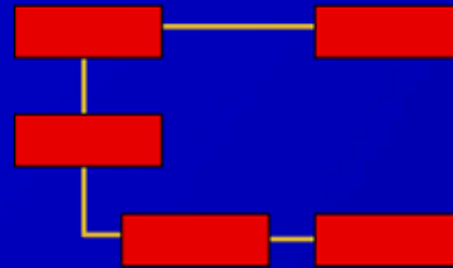


## (2): a ontologia “raciocina” sobre o pedido

---



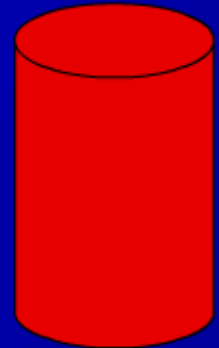
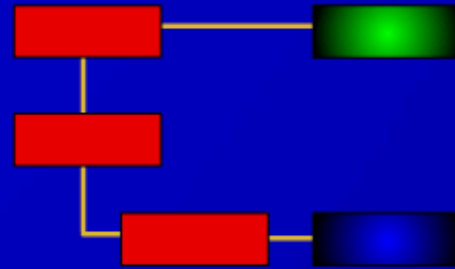
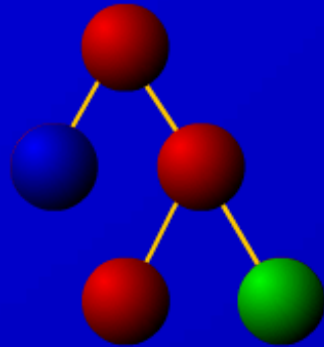
subject = bear





### (3): OWLBase converte o pedido em SQL

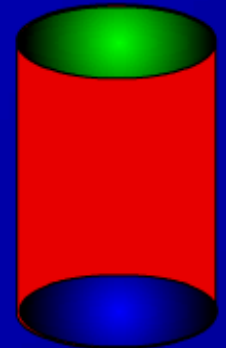
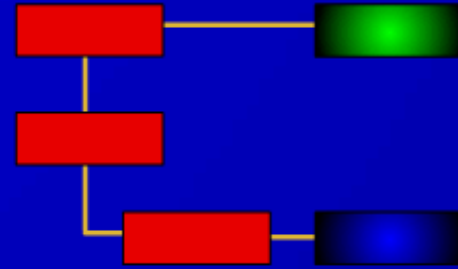
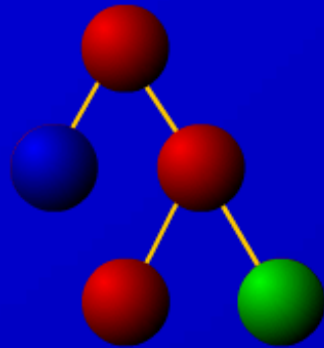
---



Image,Movie,Study...  
hasSubject  
Bear,Panda,...

(4): a metainformação é extraída da base de dados

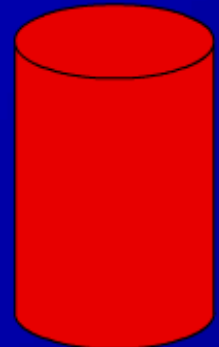
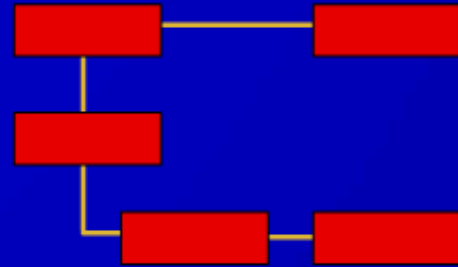
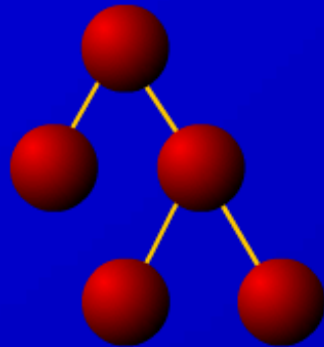
---



select \* from...

(5): a metainformação é enviada ao OWLBase em XML

---



```
<Image>  
  <Subject>  
    <Panda>  
      ....  
    </Panda>  
  </Subject>  
</Image>
```

# Em resumo

---

- Queries are made by our ontology-driven database query engine, **OWLBase**
  - OWLBase passes a query via the ImageStore ontology to the underlying PostgreSQL metadata relational database
  - The database returns metadata of studies matching the search term:
    - authors
    - title
    - description
    - network locator (URI) for the representative thumbnail image
    - IDs of all the component datasets and images
  - These XML data are then used to populate the HTML Study Results Web page that is displayed to the user
  - Many of these items link to deeper metadata
  - If the user now clicks on one of the nodes linking to deeper metadata, a new OWLBase query is initiated that returns information about that component
-



---

**Em 2012,**

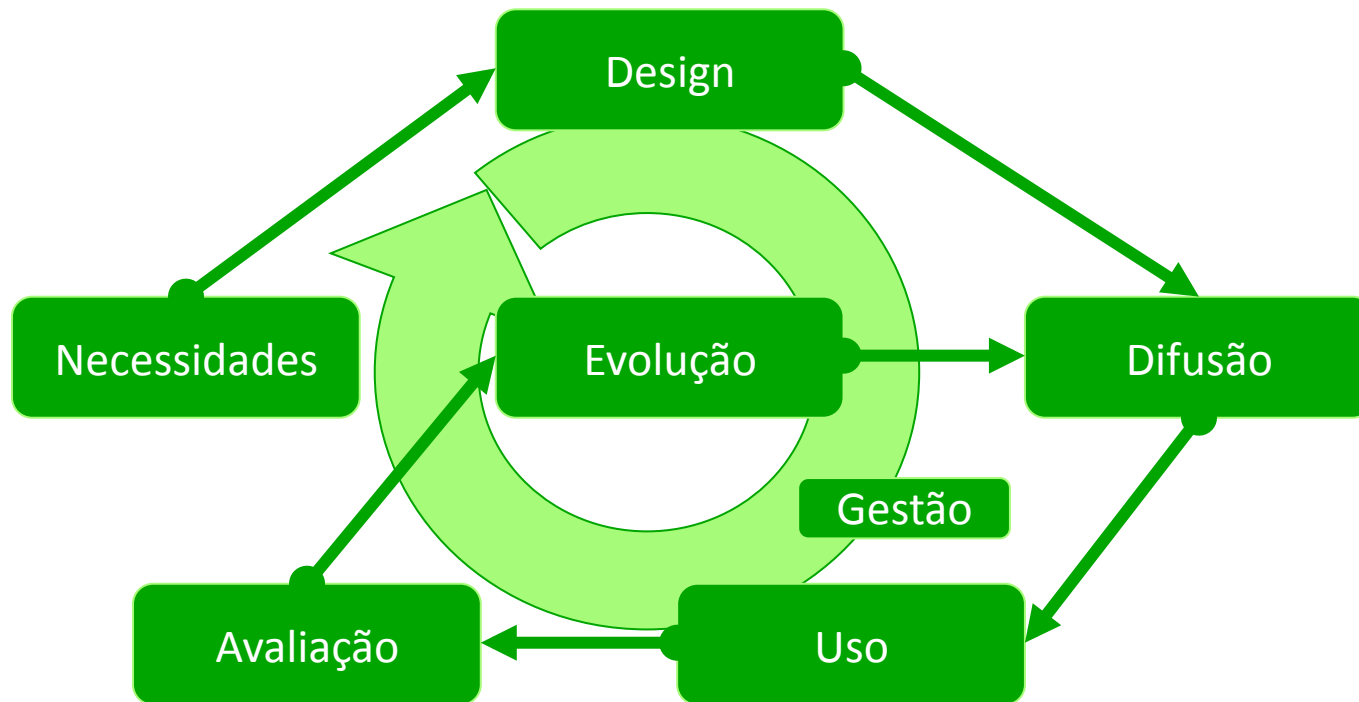
70% das páginas Web terão algum grau de anotação semântica, mas apenas 20% utilizarão tecnologias da Web Semântica extensivamente.

[Finding and Exploiting Value in Semantic Technologies on the Web  
Gartner Research Report, May 2007]



# Ciclo de Vida

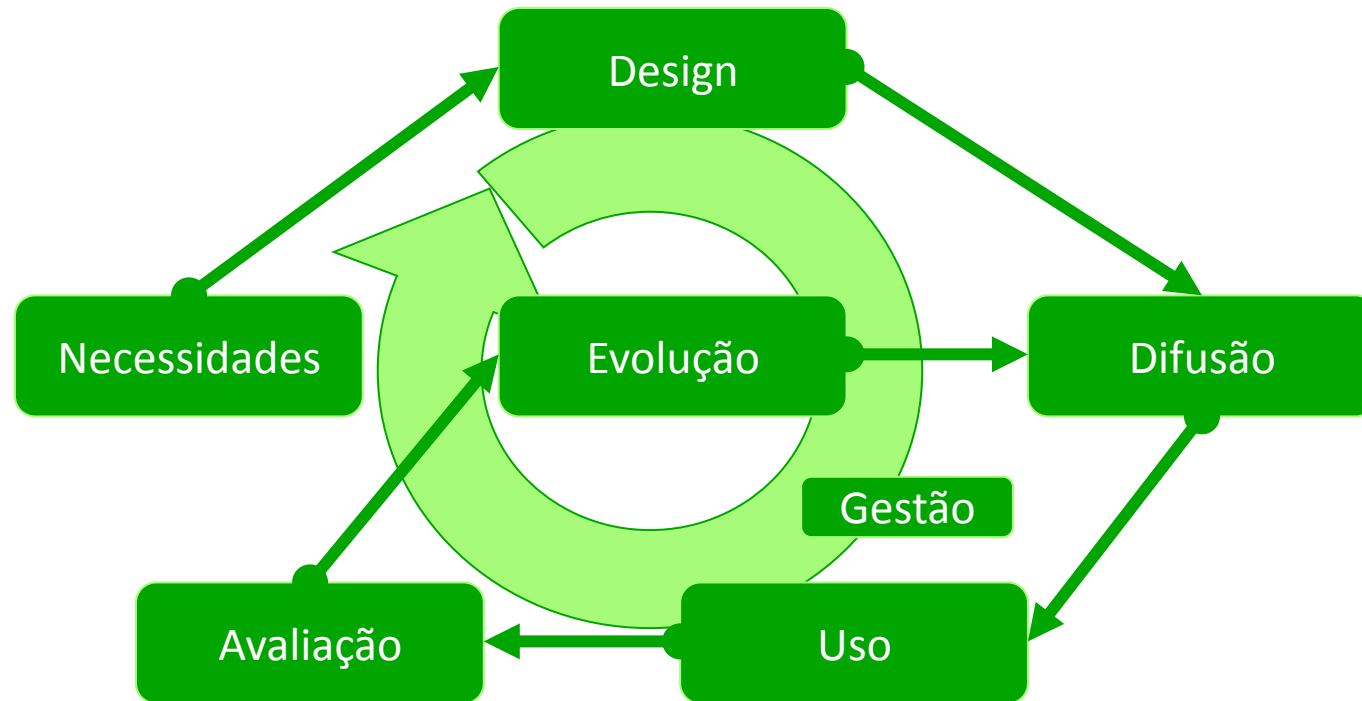
---

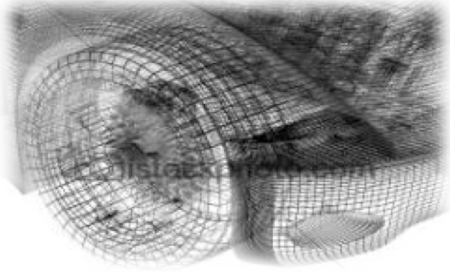




cenários motivadores, competência  
questões,...

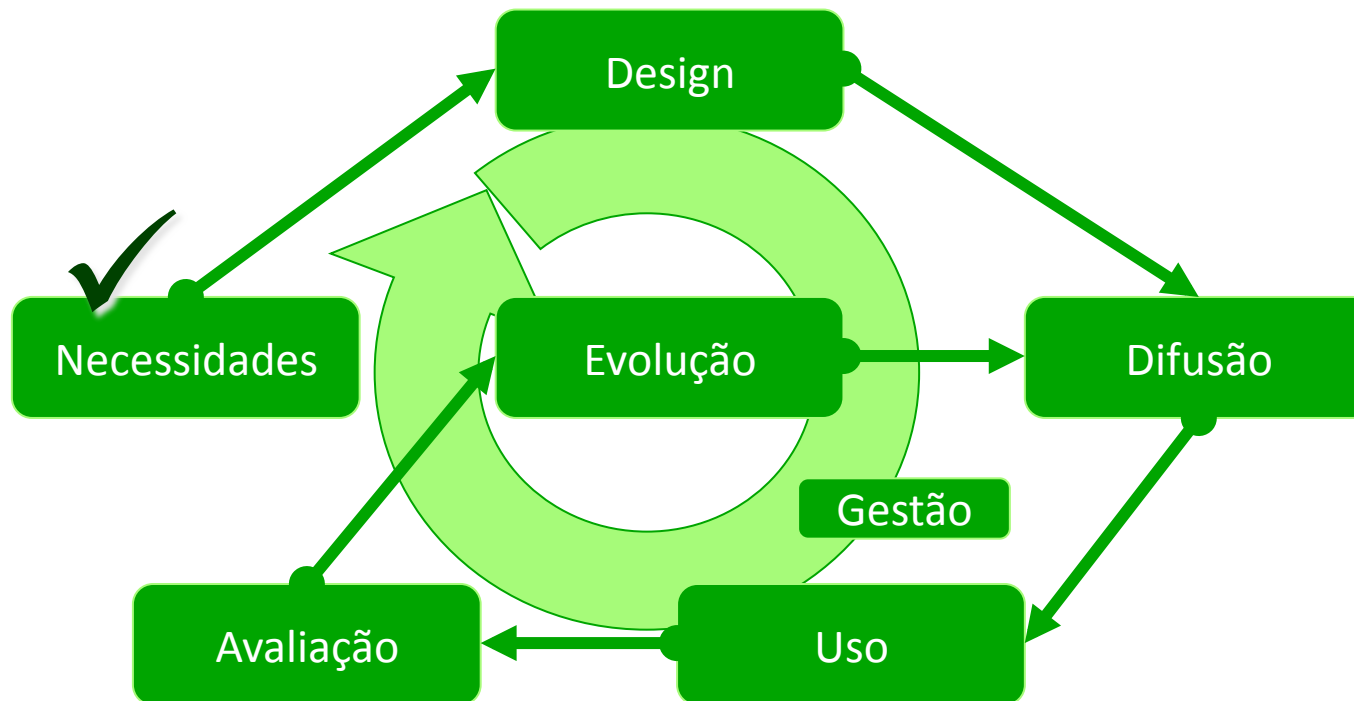
# necessidades





# design

técnicas de aquisição de conhecimento, processamento de língua natural, análise formal de conceitos, metodologias, e representações intermédias.

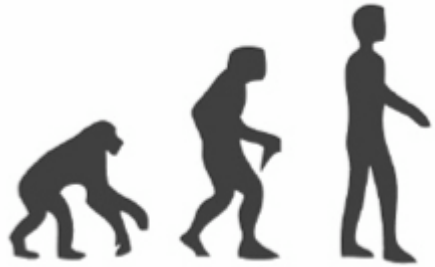






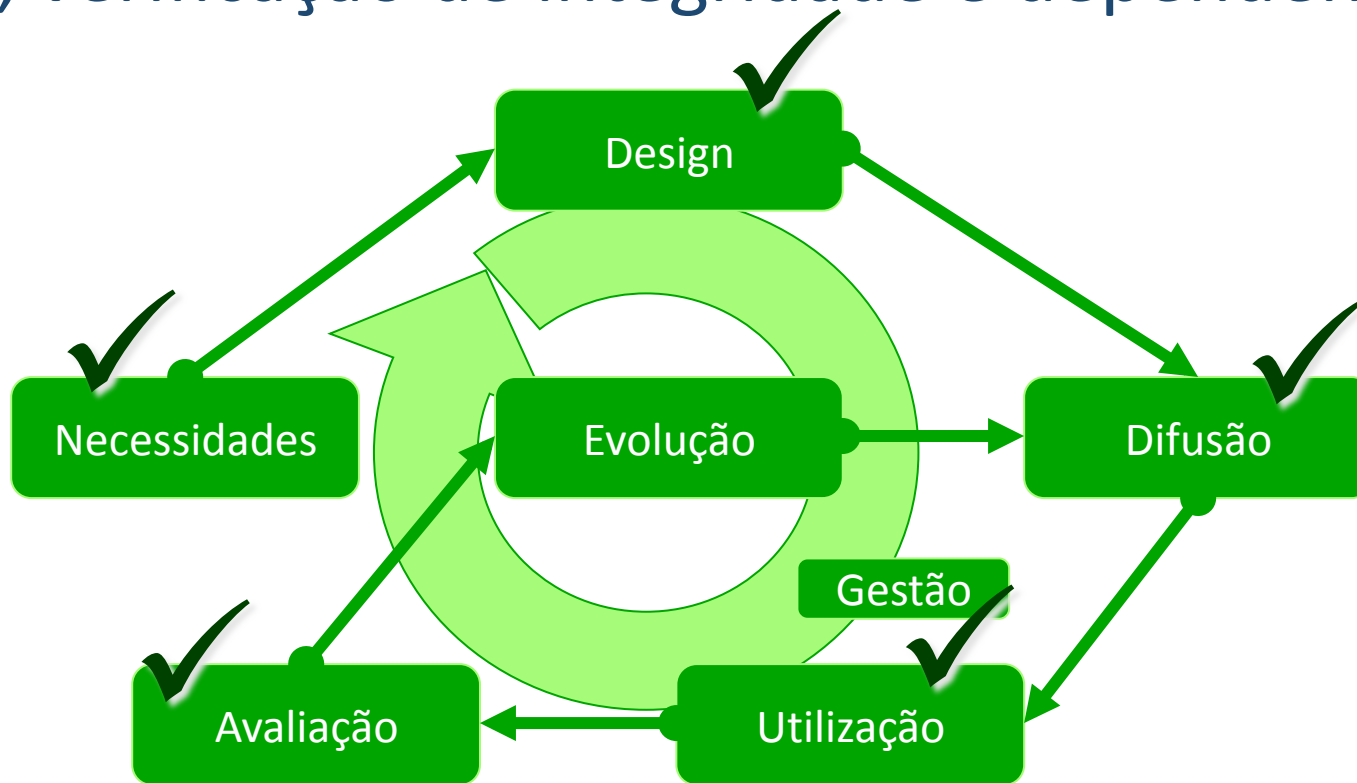






# evolução

c.f. design + gestão de versões, alinhamento de versão, verificação de integridade e dependências

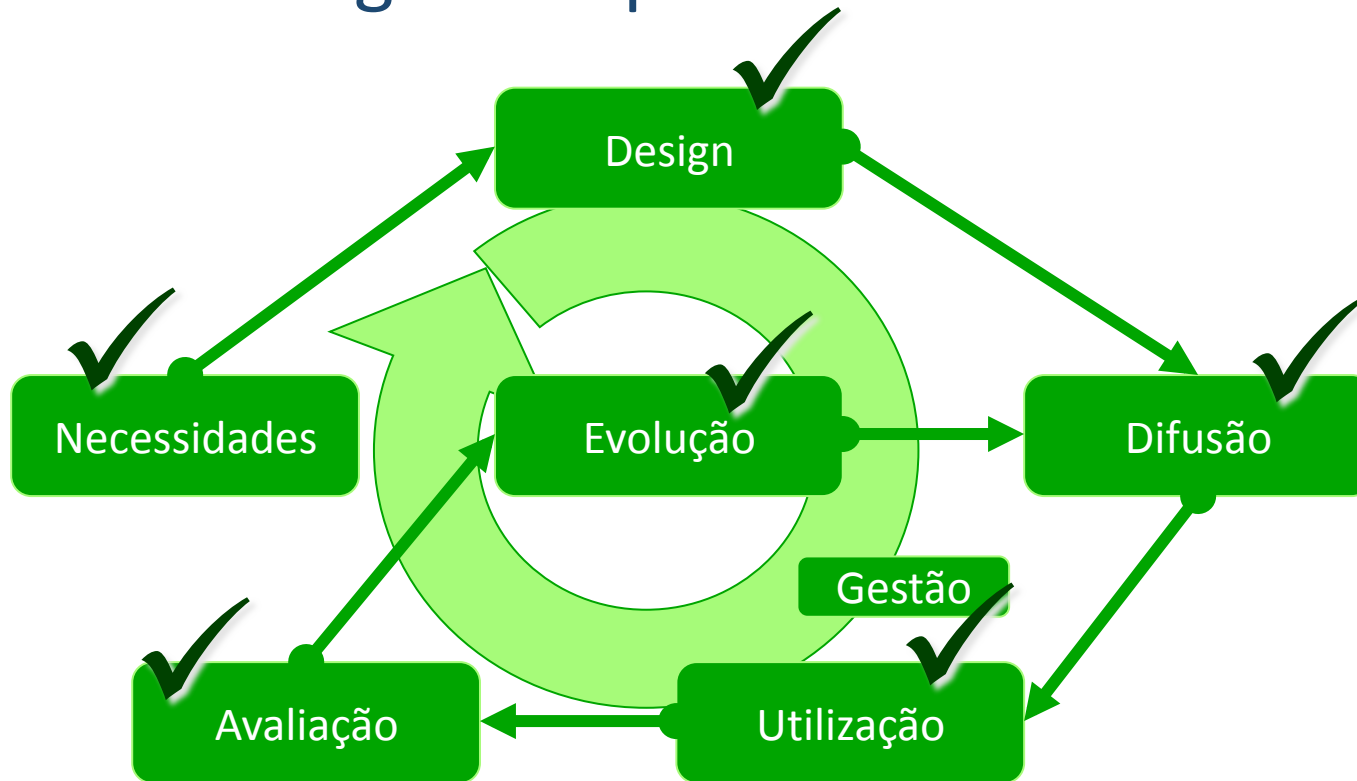




# gestão

---

como em qualquer projecto,  
uma metodologia completa



# Não há Ontologias universais

---



---

## criação & evolução bottlenecks



---

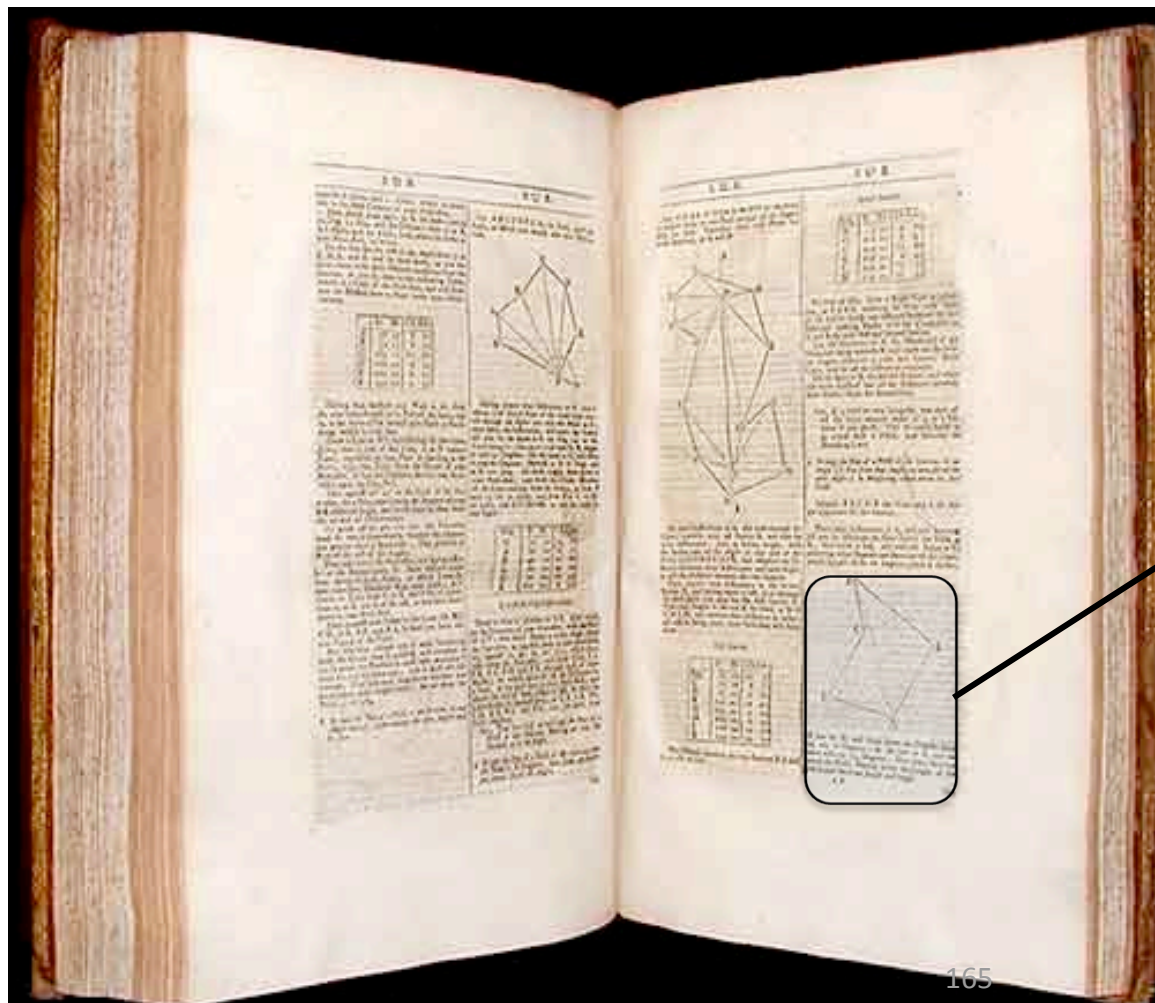
folk**S****onomies**  
in a nutshell

---



# uma anotação informação associada a um objecto

---



*origens da geometria*

---

---

# *Anotar*

*não é uma  
actividade nova*

- marcar
- descrever
- memo
- comentar
- indexar
- agrupar
- ordenar

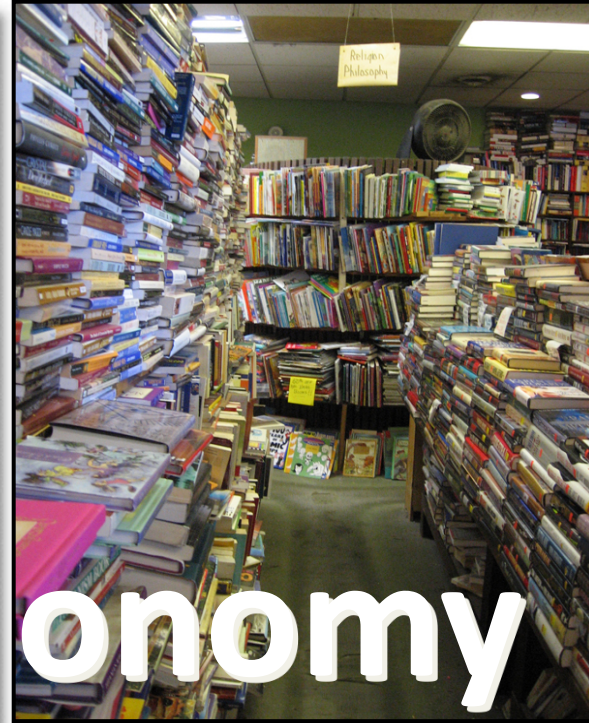
*etc.*

---



## anotação social

criar e gerir anotações  
colaborativamente para anotar e  
classificar conteúdos.



utilizar os milhões de utilizadores para anotar os milhões de Gbytes de informação



# f olksonomy

---



folks~taxonomy, um sistema de indexação de termos criado nas comunidades da internet;

É o resultado de uma anotação individual de páginas e objectos num ambiente social partilhado;

Deriva da utilização individual de vocabulários próprios com os quais se adiciona metainformação aos recursos;

Acrescenta informação aos processos cognitivos sem incrementar os custos.

[Vander Wal, 2005] [Vander Wal, 2007][Rashmi Sinha, 2005]



---

folksonomies  
não são antónimo de  
**ontologies**

---

# folksonomies

podem ser encaradas como uma maneira  
de construir e manter

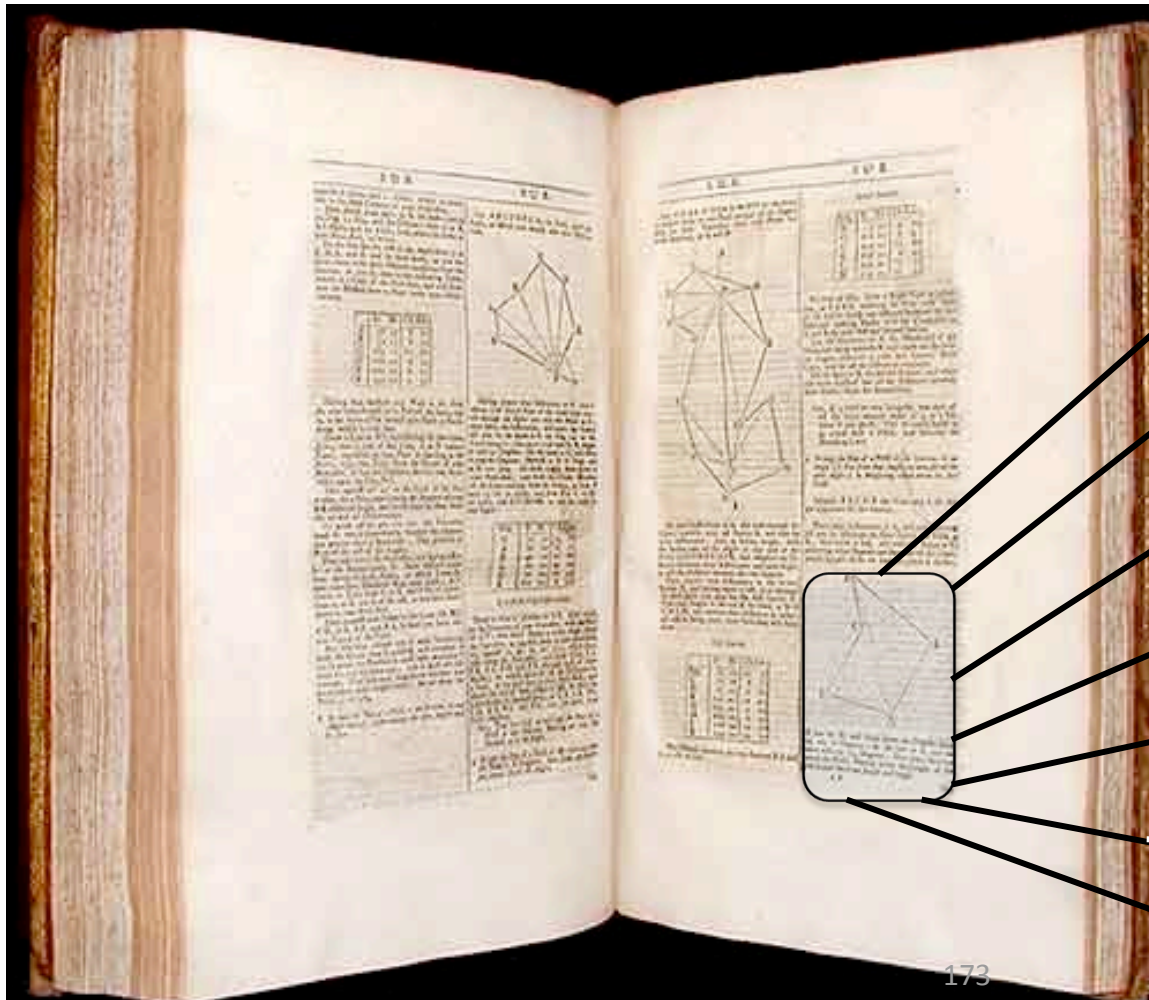
# ontologies

---



# muitas anotações para muitos usos

---



*cool*

*to compare with RR176*

*origins of geometry*

*send to Ted*

*absolument faux*

*for the SysDev team*

*;-)*

---



muitas anotações  
Voltamos ao início?

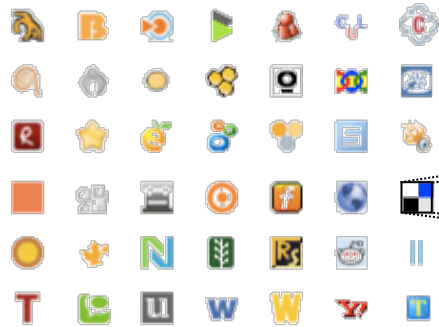


nuvem “cinzenta”

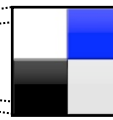


---

# *bookmarks*



my bookmarked page



socially shared bookmark



bookmark shared across people an applications

---

---

**&** ontologies  
folksonomies

---





---

“semantic web”  
*and not*

“semantic web”

[C. Welty, ISWC 2007]

---

a lightweight ontology  
allows us to do lightweight  
reasoning

[J. Hendler, ISWC 2007]





Não se pode  
prever

---

o uso futuro de  
outros







evita construir uma  
caixa negra



torna as conceptualizações  
explícitas

---



torna a tua informação  
acessível

a quem puder  
utilizá-la