

RDF e Cia

Modelo RDF, Sintaxe RDF, Esquema RDF

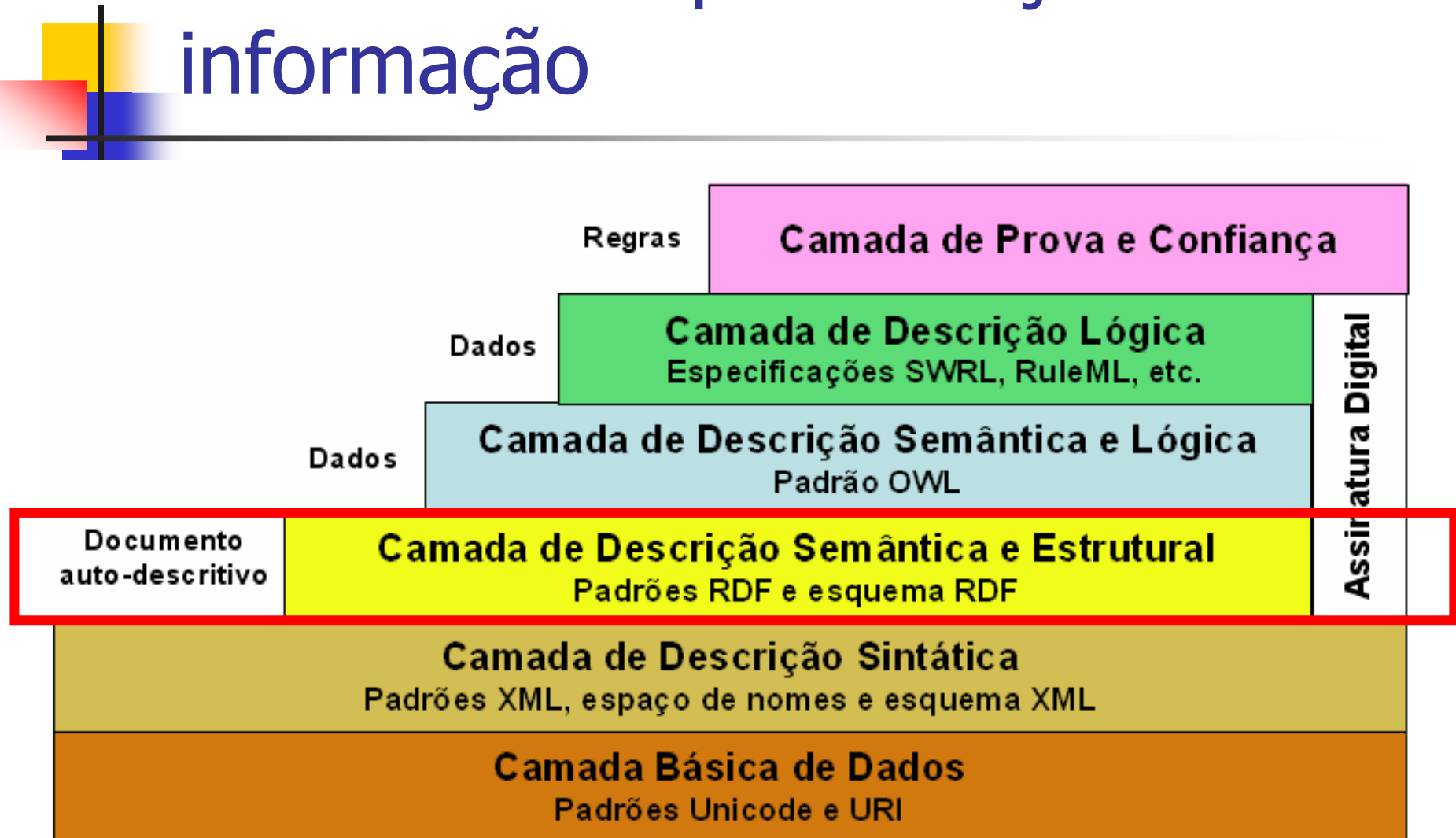
Renato Bulcão Neto
Cássio Vinicius Prazeres
Maria da Graça Pimentel

`{rbulcao,prazeres,mgp}@icmc.usp.br`



Departamento de SCC
Ciências de Computação

Padrões de representação de informação





Modelo RDF



Conceitos básicos de RDF

- Declaração
 - Tripla: sujeito – predicado – objeto
- Utiliza sintaxe XML
 - Herda os benefícios de XML
- Os principais fundamentos são
 - recursos
 - propriedades
 - declarações



Recursos

- Um objeto, ou uma “coisa”, do qual que se dizer alguma coisa
 - autores, livros, editoras, locais, pessoas, hotéis, etc.
- Todo recurso tem uma URI associada
- Uma URI pode ser
 - uma URL (endereço na Web) ou
 - algum outro tipo de identificador único



Propriedades

- Um tipo especial de recursos
 - logo são identificadas por uma URI
- Descreve relações entre outros recursos
 - Ex.: “escrito por”, “idade”, “título”, etc.



Declarações

- É a associação de um recurso a uma propriedade com determinado valor
- Tripla: sujeito – predicado – objeto
 - consiste de um recurso, uma propriedade e um valor
- Valores podem ser recursos ou literais
 - literais são valores atômicos (strings)



Três visões de uma declaração

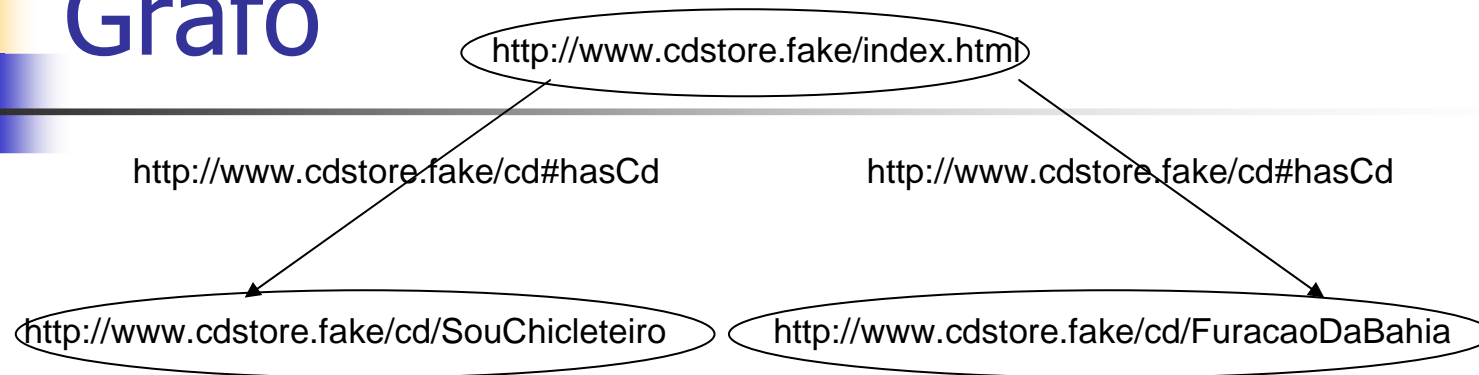
- Uma tripla
- Uma parte de um grafo
- Uma parte de um código XML
- Logo um documento RDF pode ser visto como
 - Um conjunto de triplas
 - Um grafo
 - Um documento XML



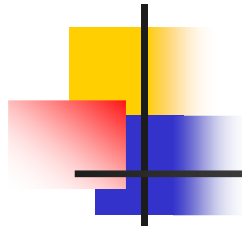
Triplas

| recurso | propriedade | valor |
|---|---|-----------------|
| http://exemplo/SouChicleteiro | http://www.cdstore.fake/cd#title | Sou Chicleteiro |
| http://exemplo/SouChicleteiro | http://www.cdstore.fake/cd#country | Brasil |

- A tripla (x, P, y) pode ser considerada como uma fórmula lógica $P(x,y)$
 - O predicado binário P relaciona o objeto x ao objeto y
 - RDF oferece apenas predicados binários (propriedades)



- Grafos direcionados com nós e arcos rotulados
 - do recurso: sujeito da declaração
 - para o valor: objeto da declaração
- O valor de uma declaração pode ser um recurso ou uma literal



Declarações RDF em XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.cdstore.fake/cd#">
  <rdf:Description rdf:about="http://www.cdstore.fake/index.html">
    <cd:hasCd rdf:resource="http://www.cdstore.fake/cd/SouChicleteiro"/>
    <cd:hasCd rdf:resource="http://www.cdstore.fake/cd/FuracaoDaBahia"/>
  </rdf:Description>
</rdf:RDF>
```

- Elemento `rdf:Description` faz declaração sobre o recurso `http://www.cdstore.fake/index.html`
- Dentro de descrição
 - aparecem as marcas referentes às propriedades



Tipos de dados (1)

- Literais tipadas podem ser usadas, caso seja necessário

(<http://www.cdstore.fake/cd/SouChicleteiro>,
<http://www.cdstore.fake/cd#price>,
"10.90"^^<http://www.w3.org/2001/XMLSchema#double>)



Tipos de dados (2)

- a notação ^^ indica o tipo de uma literal
- Os tipos de dados mais utilizados são aqueles definidos na linguagem Esquema XML
 - Ex: boolean, int, float, time, date, etc.

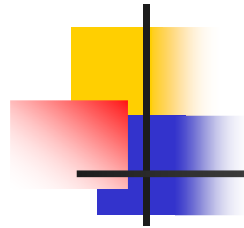


Sintaxe RDF



Declarações em XML

- Grafos são ferramentas poderosas para o entendimento por humanos
 - Mas a Web Semântica requer representações que possam ser acessadas e processadas por máquinas
- RDF/XML
 - XML não é parte do modelo de dados RDF
 - Apenas uma forma de serialização do modelo



Declarações em XML

- Um documento RDF é representado por um elemento XML com a marca `rdf:RDF`
- O conteúdo desse elemento é um conjunto de descrições: marcas `rdf:Description`
- Cada descrição faz uma declaração sobre um recurso identificado de três formas
 - atributo `about`: referencia um recurso existente
 - atributo `ID`: cria um novo recurso
 - sem URI: recurso anônimo



Cursos de universidade: Exemplo

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:uni="http://www.mydomain.org/uni-ns">
  <rdf:Description rdf:about="949318">
    <uni:name>David Billington</uni:name>
    <uni:title>Associate Professor</uni:title>
    <uni:age rdf:datatype="&xsd:integer">27</uni:age>
  </rdf:Description>
  <rdf:Description rdf:about="CIT1111">
    <uni:courseName>Discrete Maths</uni:courseName>
    <uni:isTaughtBy>David Billington</uni:isTaughtBy>
  </rdf:Description>
  <rdf:Description rdf:about="CIT2112">
    <uni:courseName>Programming III</uni:courseName>
    <uni:isTaughtBy>Michael Maher</uni:isTaughtBy>
  </rdf:Description>
</rdf:RDF>
```



rdf:about e rdf:ID

- Um elemento `rdf:Description` tem
 - Um atributo `rdf:about` indicando que o recurso já foi definido em outro lugar
 - Um atributo `rdf:ID` indicando que o recurso está sendo definido

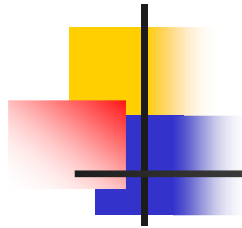


Elementos de Propriedades

- Conteúdo dos elementos `rdf:Description`

```
<rdf:Description rdf:about="CIT3116">  
  <uni:courseName>Knowledge  
  Representation</uni:courseName>  
  <uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>  
</rdf:Description>
```

- `uni:courseName` e `uni:isTaughtBy` definem dois pares propriedade-valor para CIT3116 (duas declarações RDF)



Tipos de dados (1)

- O atributo `rdf:datatype="&xsd:integer"` é usado para indicar o tipo de dados do valor da propriedade `age`

```
<rdf:Description rdf:about="949318">  
  <uni:name>David Billington</uni:name>  
  <uni:title>Associate Professor</uni:title>  
  <uni:age rdf:datatype="&xsd:integer">27</uni:age>  
</rdf:Description>
```



Tipos de dados (2)

- A propriedade age tem sido definida como do tipo "&xsd:integer"
 - É necessário indicar o tipo do valor dessa propriedade toda vez que ela é utilizada
 - Isto garante que um processador RDF possa atribuir o tipo correto da propriedade mesmo se não encontrar uma definição anterior
 - Este cenário é muito comum de ocorrer na irrestrita WWW



Atributo rdf:resource (1)

- Relacionamentos entre cursos e professores não foram formalmente definidos
 - Existiam implicitamente através do uso do mesmo nome do professor
- O uso do mesmo nome pode ser apenas uma coincidência para uma aplicação computacional
- Pode-se denotar que duas entidades são a mesma usando o atributo rdf:resource



Atributo rdf:resource (2)

```
<rdf:Description rdf:about="CIT1111">  
  <uni:courseName>Discrete Mathematics</uni:courseName>  
  <uni:isTaughtBy rdf:resource="949318"/>  
</rdf:Description>  
<rdf:Description rdf:about="949318">  
  <uni:name>David Billington</uni:name>  
  <uni:title>Associate Professor</uni:title>  
</rdf:Description>
```



Referenciando recursos definidos externamente

- Ex: para referenciar o recurso CIT1111 definido externamente:
 - `http://www.mydomain.org/uni-ns#CIT1111` como o valor de `rdf:about`
 - `www.mydomain.org/uni-ns` é a URI onde se encontra a definição de CIT1111
- A descrição com um ID define um fragmento da URI que pode ser usado para referenciar a descrição



Descrições aninhadas

- Descrições podem ser definidas dentro de outras descrições
- Outros cursos, tal como CIT3112, podem continuar a referenciar um novo recurso com o ID 949318
- Mesmo que uma descrição seja definida dentro de outra descrição o seu escopo é global



Estruturando documentos RDF usando o elemento rdf:type

```
<rdf:Description rdf:ID="CIT1111">  
  <rdf:type rdf:resource="http://www.mydomain.org/uni-ns#course"/>  
  <uni:courseName>Discrete Maths</uni:courseName>  
  <uni:isTaughtBy rdf:resource="#949318"/>  
</rdf:Description>
```

```
<rdf:Description rdf:ID="949318">  
  <rdf:type rdf:resource="http://www.mydomain.org/uni-ns#lecturer"/>  
  <uni:name>David Billington</uni:name>  
  <uni:title>Associate Professor</uni:title>  
</rdf:Description>
```



Sintaxe abreviada (1)

- Regras de simplificação
 - Propriedades sem elementos filhos e dentro de elementos de descrição podem ser substituídas por atributos XML
 - Para elementos de descrição com um elemento de tipificação pode-se usar o nome especificado em `rdf:type` em vez de `rdf:Description`



Sintaxe abreviada (2)

- Estas regras criam variações sintáticas da mesma declaração RDF
 - São equivalentes de acordo com o modelo de dados RDF, porém têm diferentes sintaxes XML



Sintaxe abreviada: Exemplo (1)

```
<rdf:Description rdf:ID="CIT1111">  
  <rdf:type rdf:resource="http://www.mydomain.org/uni-ns#course"/>  
  <uni:courseName>Discrete Maths</uni:courseName>  
  <uni:isTaughtBy rdf:resource="#949318"/>  
</rdf:Description>
```



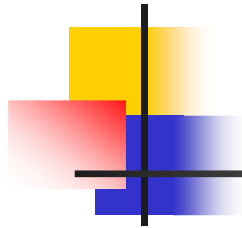
Sintaxe abreviada: Exemplo (2)

- Aplicação da primeira regra de simplificação

```
<rdf:Description rdf:ID="CIT1111" uni:courseName="Discrete Maths">  
  <rdf:type rdf:resource="http://www.mydomain.org/uni-ns#course"/>  
  <uni:isTaughtBy rdf:resource="#949318"/>  
</rdf:Description>
```

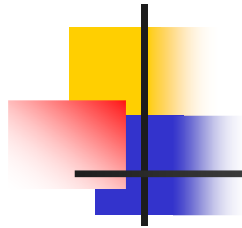
- Aplicação da segunda regra de simplificação

```
<uni:course rdf:ID="CIT1111" uni:courseName="Discrete Maths">  
  <uni:isTaughtBy rdf:resource="#949318"/>  
</uni:course>
```



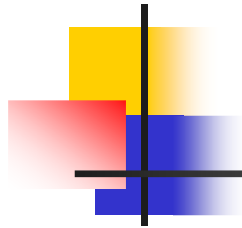
Elementos Container

- Contém um número de recursos ou atributos sobre algo que se deseja fazer declarações como um todo
- Ex: pode-se querer declarar algo sobre os cursos de um determinado professor
- Os conteúdos de um elemento container são nomeados `rdf:_1`, `rdf:_2`, etc.



Três tipos de elementos container

- `rdf:Bag` é um container sem ordem associada, permitindo múltiplas ocorrências
 - Ex: documentos em uma pasta
- `rdf:Seq` é um container ordenado que pode conter múltiplas ocorrências
 - Ex: módulos de um curso, itens em uma agenda
- `rdf:Alt` é um conjunto de alternativas
 - Ex: traduções de um documento em várias línguas



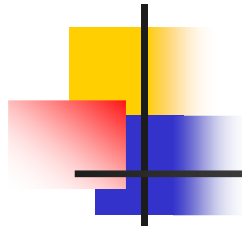
Exemplo de container Bag

```
<uni:lecturer rdf:ID="949352" uni:name="Grigoris Antoniou"
               uni:title="Professor">
  <uni:coursesTaught>
    <rdf:Bag>
      <rdf:_1 rdf:resource="#CIT1112"/>
      <rdf:_2 rdf:resource="#CIT3116"/>
    </rdf:Bag>
  </uni:coursesTaught>
</uni:lecturer>
```



Exemplo de container Alt

```
<uni:course rdf:ID="CIT1111"
  uni:courseName="Discrete Mathematics">
  <uni:lecturer>
    <rdf:Alt>
      <rdf:li rdf:resource="#949352"/>
      <rdf:li rdf:resource="#949318"/>
    </rdf:Alt>
  </uni:lecturer>
</uni:course>
```



Container com atributo rdf:ID

```
<uni:lecturer rdf:ID="949318"
               uni:name="David Billington">
  <uni:coursesTaught>
    <rdf:Bag rdf:ID="DBcourses">
      <rdf:_1 rdf:resource="#CIT1111"/>
      <rdf:_2 rdf:resource="#CIT3112"/>
    </rdf:Bag>
  </uni:coursesTaught>
</uni:lecturer>
```



Esquema RDF



Idéias básicas de Esquema RDF

- RDF permite que usuários descrevam recursos em seus próprios vocabulários
 - RDF não assume ou não define semântica de de um domínio em particular
- Para isto, pode-se usar o Esquema RDF que define:
 - Classes e Propriedades
 - Herança e Hierarquia de classes
 - Hierarquia de propriedades



Classes

- É necessário distinguir entre
 - “Coisas” concretas (objetos individuais) e o domínio
 - Conjuntos de individuais compartilham propriedades chamadas de classes: lojas de cd, cds, músicas, artistas, compositores, etc.



Classes e suas instâncias

- Objetos individuais que pertencem a uma classe são referidos como uma instância da classe
- O relacionamento entre instâncias e classes em RDF se dá através de `rdf:type`



Porque classes são úteis em RDF

- Impor restrições em o que pode ser declarado em um documento RDF
 - Como em linguagens de programação
 - Ex: $A+1$, onde A é um Array
 - Não permitir declarações sem sentido
- Exemplos de classes de uma loja de CDs
 - Artist, MusicCD, CDCatalog, Company, Song, Composer e Country



Exemplos de declarações sem sentido não permitidas com o uso de classes

- “100% Você” é cantada pela “Sony Music”
 - As músicas devem ser cantadas apenas pelos artistas
 - Restrição no valor da propriedade “é cantada pela” (restrição de range)
- “Sony Music” compôs “Mulher Amada”
 - Apenas compositores devem compor músicas
 - Restrição no objeto em que a propriedade está sendo aplicada (restrição de domínio)

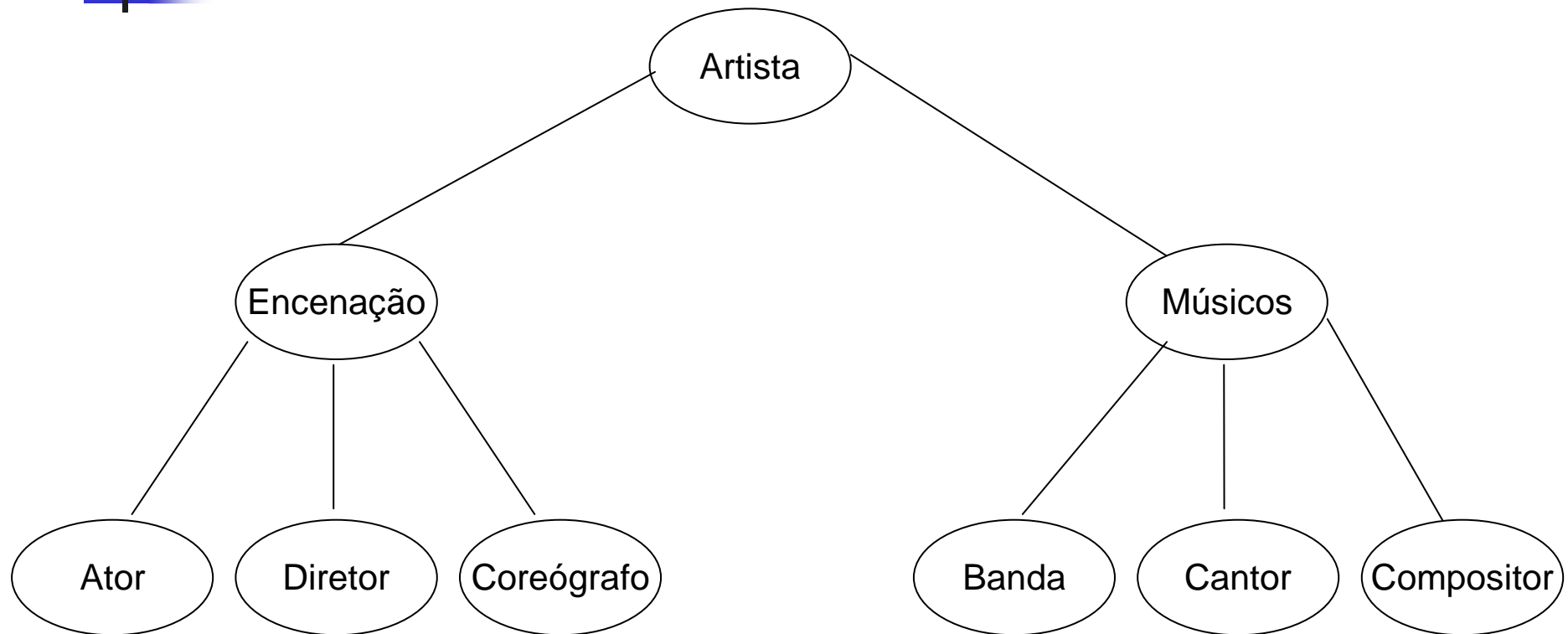


Hierarquia de classes

- Classes podem ser organizadas em hierarquias
 - A é uma sub-classe de B se toda instância de A é também uma instância de B
 - Então, B é uma super-classe de A
- Um grafo de sub-classes não precisa ser uma árvore
- Uma classe pode ter muitas super-classes



Hierarquia de classes: Exemplo





Herança na hierarquia de classes

- Restrição de range: apenas músicos participam de CDs
- Bel Marques é um cantor
- Ele herda, de Músicos, a habilidade de poder participar de CDs
- Isto é feito em RDF Schema fixando a semântica de “é sub-classe de”



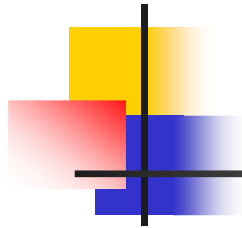
Hierarquia de propriedades

- Relacionamentos hierárquicos para propriedades
 - Ex: “é cantada por” é uma sub-propriedade de “é interpretada por”
 - Se uma música M é cantada por um cantor C, então M também é interpretada por C



Esquema RDF escrita em RDF

- As primitivas de Esquema RDF são definidas usando recursos e propriedades
- Para declarar que um “cantor” é uma sub-classe de “músico”
 - Define-se os recursos cantor, músico e subClassOf
 - Define a propriedade subClassOf
 - Escreve a tripla (subClassOf,cantor,músico)
- Pode-se utilizar a sintaxe RDF/XML



Principais classes

- `rdfs:Resource` – a classe de todos os recursos
- `rdfs:Class` – a classe de todas as classes
- `rdfs:Literal` – a classe de todas as literais (strings)
- `rdf:Property` – a classe de todas as propriedades
- `rdf:Statement` – a classe de todas as declarações efetuadas



Principais Propriedades (1)

- `rdf:type` relaciona um recurso à sua classe
 - O recurso é declarado como sendo uma instância da classe
- `rdfs:subClassOf` relaciona uma classe com uma de suas super-classes
 - Todas as instâncias de uma classe são instâncias de sua super-classe
- `rdfs:subPropertyOf` relaciona propriedade e suas sub-propriedades



Principais Propriedades (2)

- `rdfs:domain` especifica o domínio de uma propriedade
 - A classe dos recursos que podem aparecer como sujeito em um predicado
 - Se o domínio não for especificado qualquer recurso pode ser o sujeito da declaração
- `rdfs:range` especifica a imagem de uma propriedade
 - A classe dos recursos que podem aparecer como valores de um predicado



Exemplos

```
<rdfs:Class rdf:about="#Composer">  
  <rdfs:subClassOf rdf:resource="#Artist"/>  
</rdfs:Class>
```

```
<rdf:Property rdf:ID="cdTitle">  
  <rdfs:range  
    rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
  <rdfs:domain rdf:resource="#MusicCD"/>  
</rdf:Property>
```



Relacionamentos entre as principais classes e propriedades

- `rdfs:subClassOf` e `rdfs:subPropertyOf` são, por definição, transitivas
- `rdfs:Class` é uma sub-classe de `rdfs:Resource`
 - Toda classe é um recurso
- `rdfs:Resource` é uma instância de `rdfs:Class`
 - `rdfs:Resource` é a classe de todos os recursos, assim, é uma classe
- Toda classe é uma instância de `rdfs:Class`



Propriedades utilitárias

- `rdfs:seeAlso` relaciona um recurso com outro recurso que o explica
- `rdfs:comment` são comentários (geralmente textos) que podem ser associados com um recurso
- `rdfs:label` é um nome, legível para humanos, que pode ser associado a um recurso



Exemplo

```
<rdfs:Class rdf:ID="musico">  
  <rdfs:comment>  
    A classe dos músicos. Todos os músicos são artistas.  
  </rdfs:comment>  
  <rdfs:subClassOf rdf:resource="#artista"/>  
</rdfs:Class>
```



Pontos para discussão

- Esquema RDF é bastante primitiva como linguagem de modelagem para Web
- Muitas primitivas de modelagem não são tratadas por Esquema RDF
- Portanto, existe uma necessidade de camada de ontologias sobre a tríade modelo, sintaxe e esquema RDF

RDF e Cia

Modelo RDF, Sintaxe RDF, Esquema RDF

Renato Bulcão Neto

Cássio Vinicius Prazeres

Maria da Graça Pimentel

`{rbulcao,prazeres,mgp}@icmc.usp.br`



Departamento de SCC
Ciências de Computação