Semantic Web Services

Henrique Moniz

Sumário

1. Enquadramento

2. Viagem pela Semantic Web
   - XML
   - RDF
   - DAML+OIL

3. Semantic Web Services e DAML-S

4. Conclusões

Enquadramento

- A finalidade dos Web Services é proporcionar a infraestrutura para interacções B2B
- As tecnologias realmente estabelecidas são WSDL e o SOAP. O UDDI está num estado maduro mas a sua utilização ainda tem sido limitada
- O avanço trazido por estas tecnologias ainda não é suficiente para alcançar O Santo Graal

O Santo Graal

- Interacção dinâmica numa comunidade de negócios completamente aberta é vista como o Santo Graal dos Web services.

- Clientes automatizados fariam o *browsing* de registos UDDI, descobririam os serviços adequados, como interagir com os serviços e, finalmente, invocariam os serviços, tudo programaticamente.

Semantic Web

- Através da *Semantic Web* tenta-se alcançar uma automatização completa de todas as fases dos Web services.

- Normalizar a representação e o manuseamento de todos os metadados usados para descrever os Web services e todo os aspectos relacionados com a sua utilização

Semantic Web

- O conceito-chave é Ontologia.
- O objectivo é proporcionar uma compreensão comum e partilhada de um dado domínio da realidade.
- Tipicamento será uma estrutura de dados hierararquizada que contenha todas as entidades e as relações e regras entre elas, num dado domínio.

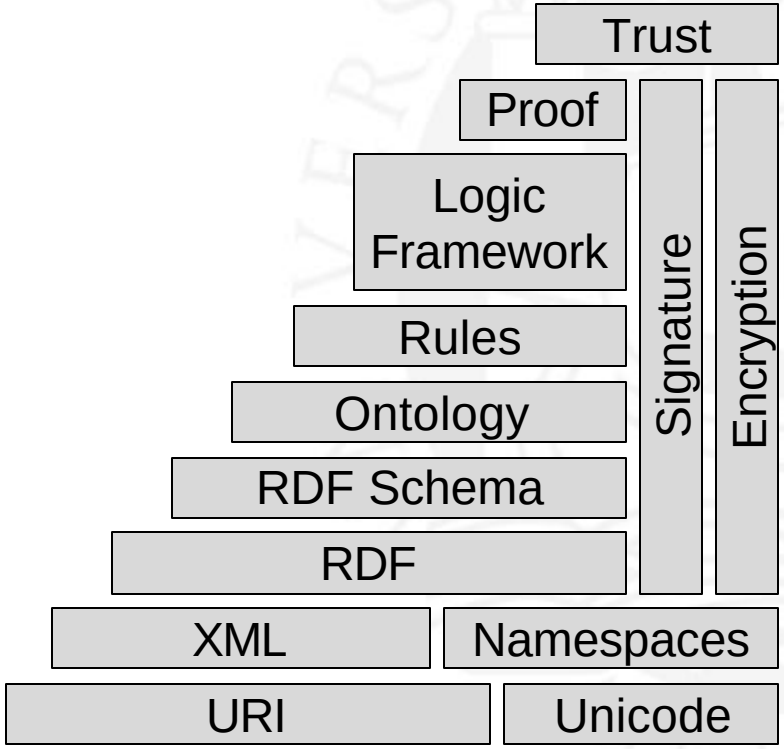
Semantic Web

- Uma ontologia permitiria a uma máquina uma compreensão semântica sobre o domínio da ontologia

- Se chegarmos a uma ontologia sobre Web services, conseguiremos então automatizar todos os processos que envolvem o ciclo de vida de um Web service

Semantic Web

- Semantic Web Stack



Semantic Web

- URI

• Uniform Resource Identifier

• Estão na fundação da Web

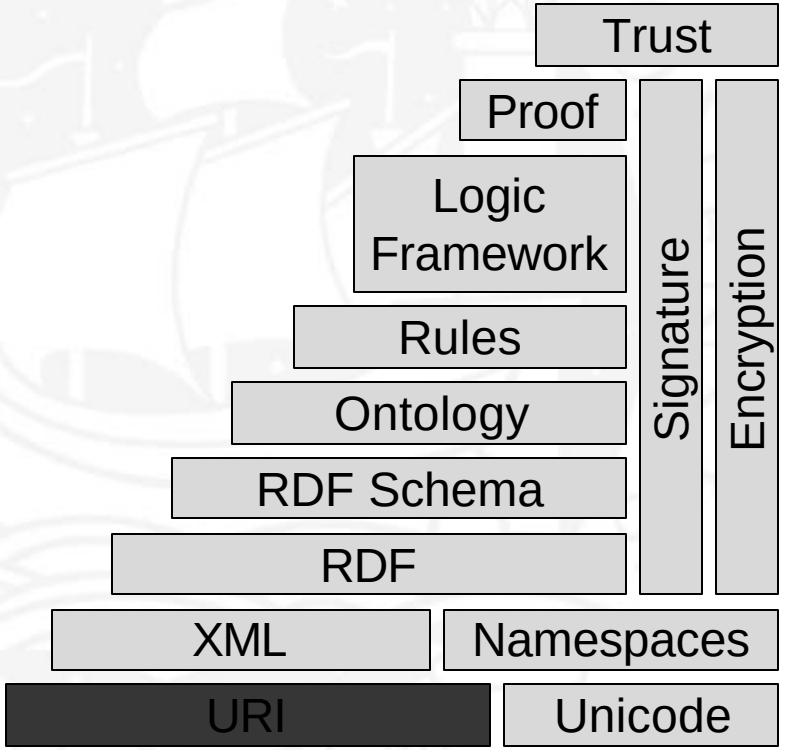
• Identificam univocamente items (ou recursos) na Web

• Exemplos:

  • http://osc.di.fc.ul.pt/tm

  • mailto:tm@di.fc.ul.pt

• Qualquer coisa pode ser identificada na Web por um URI (documentos, pessoas, etc.)



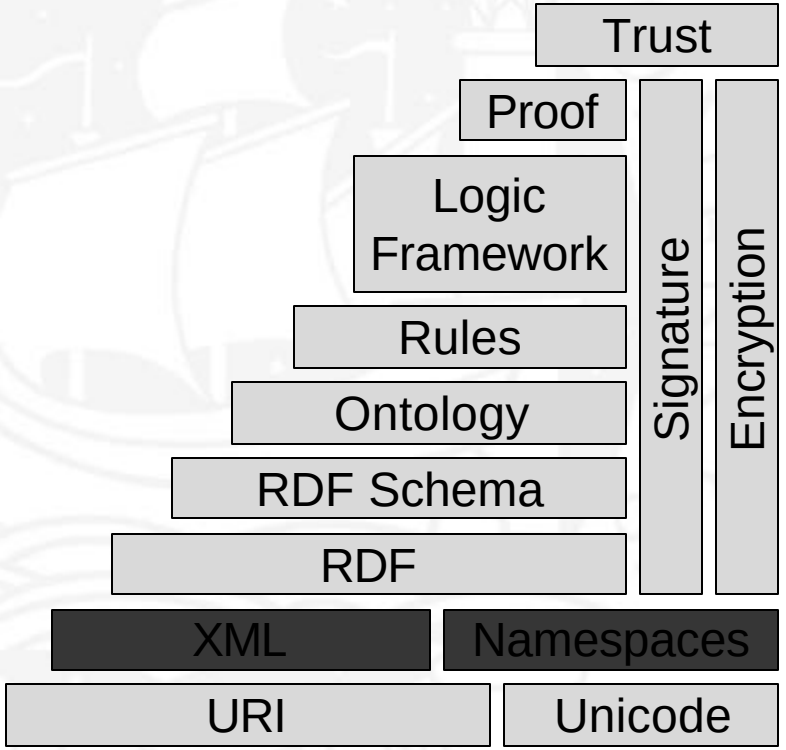
Semantic Web

- XML & Namespaces

• Facilitar a partilha de documentos através da Web

• Permite definir uma sintaxe estrita e comum de documentos de modo a que os algoritmos de *parsing* sejam simples, rápidos e eficientes

• Anexa metadados a partes (internas) do documento



Semantic Web

- XML & Namespaces

Manel tem um cão.

Documento de texto

```
<sentence>
<person href=http://manel.com>Manel</person> tem um <animal>cão</animal>.
</sentence>
```

Documento XML

```
<sentence
    xmlns="http://example.org/xml/documents/"
    xmlns:c="http://animals.example.net/xmlns/"
><c:person c:href="http://manel.com">Manel</c:person> tem um
<c:animal>cão</c:animal>.
</sentence>
```

Documento XML com Namespaces

Semantic Web

- RDF – Resource Description Framework

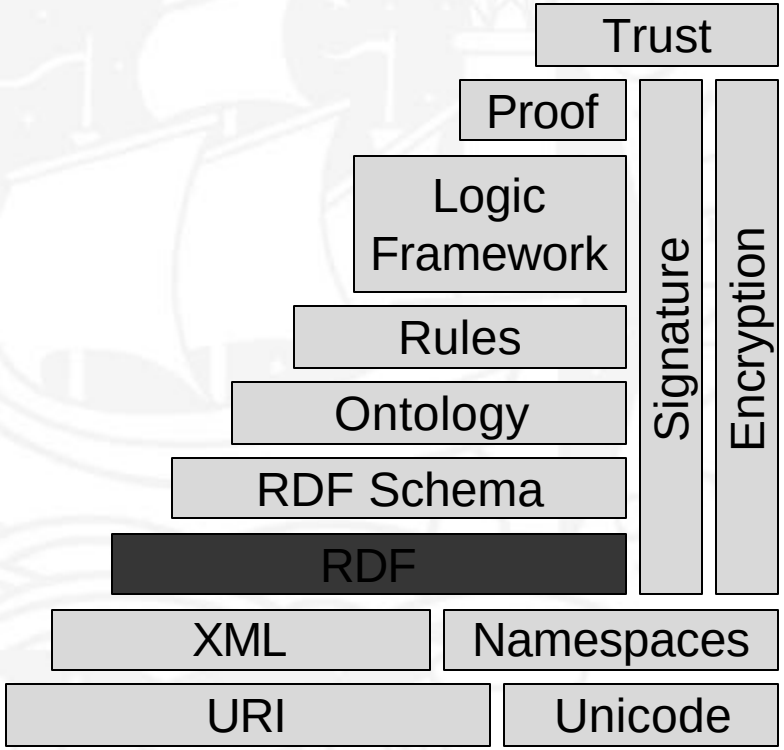
• Proporciona uma forma de criar declarações que são *machine-processable*

• As declarações RDF são como frases simples: [Predicado] [Sujeito] [Complemento directo]

• Anexa metadados ao documento como uma entidade singular

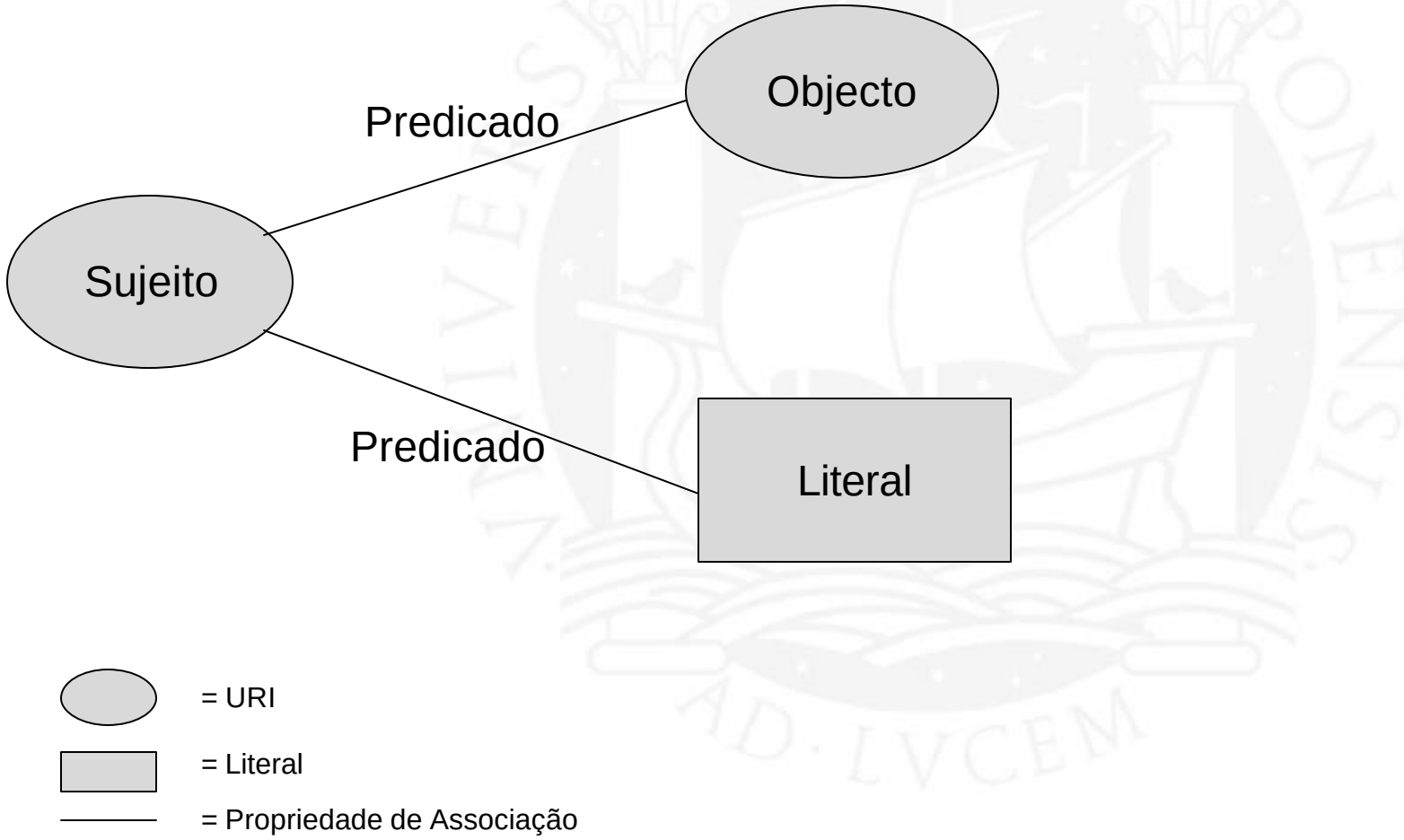
• Ao contrário de XML, em vez de anexar metadados a secções “internas” do documento, RDF cria metadados “externamente”, como o autor, data de criação, etc

• Já permite construir ontologias simples



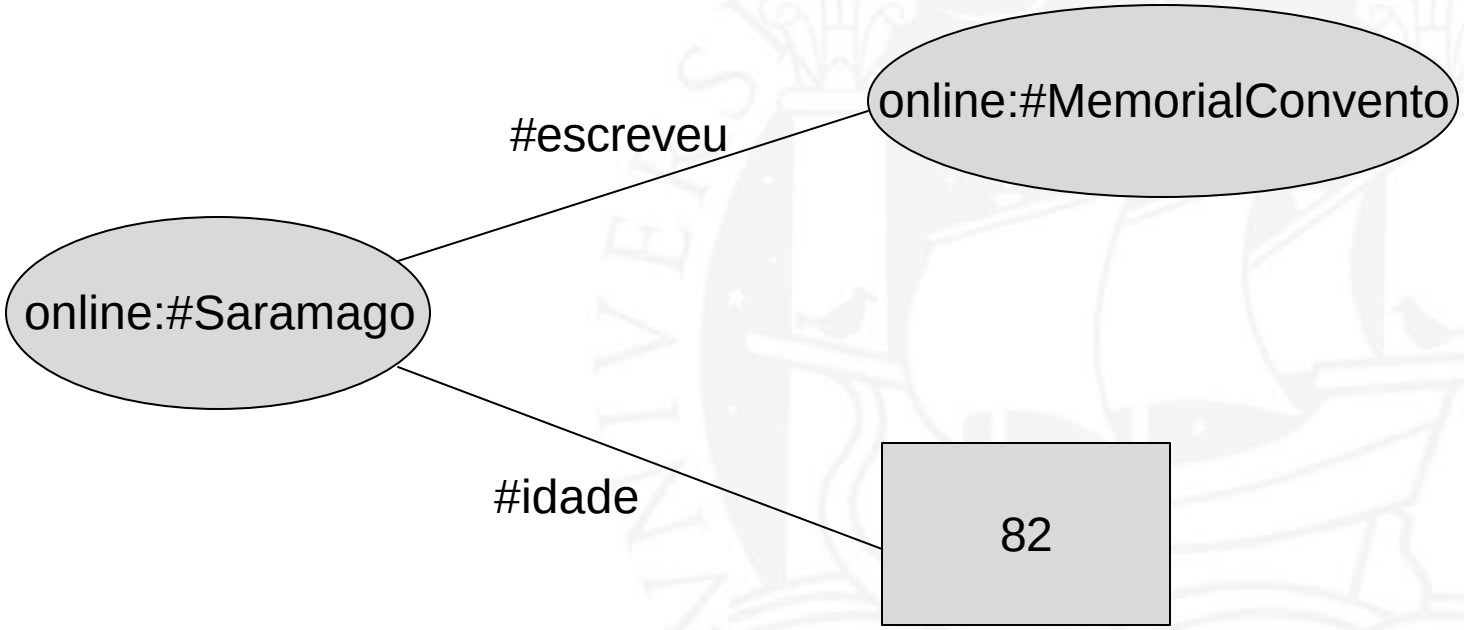
Semantic Web

- RDF – Resource Description Framework



Semantic Web

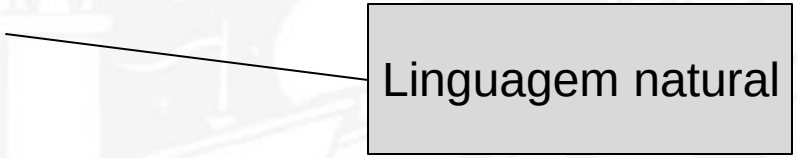
- RDF – Resource Description Framework



Semantic Web

- RDF – Resource Description Framework

José Saramago escreveu o Memorial do Convento.
José Saramago tem 82 anos.



<http://saramago.com> <http://autor.example.org/terms/Escreveu>
<http://www.books.org/MemorialConvento>

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:autor="http://autor.example.org/terms/"
>
   <rdf:Description rdf:about="http://saramago.com">
      <autor:Escreveu rdf:resource="http://www.books.org/MemorialConvento"/>
      <pessoas:idade>82</pessoas:idade>
   </rdf:Description>
</rdf:RDF>
```

Semantic Web

- RDF – Resource Description Framework

José Saramago escreveu o Memorial do Convento.
José Saramago tem 82 anos.

<http://saramago.com> <http://autor.example.org/terms/Escreveu>
<http://www.books.org/MemorialConvento>

Triplas RDF

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:autor="http://autor.example.org/terms/"
>
   <rdf:Description rdf:about="http://saramago.com">
      <autor:Escreveu rdf:resource="http://www.books.org/MemorialConvento"/>
      <pessoas:idade>82</pessoas:idade>
   </rdf:Description>
</rdf:RDF>
```

Semantic Web

- RDF – Resource Description Framework

José Saramago escreveu o Memorial do Convento.
José Saramago tem 82 anos.

<http://saramago.com> <http://autor.example.org/terms/Escreveu>
<http://www.books.org/MemorialConvento>

Serialização XML

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:autor="http://autor.example.org/terms/"
>
   <rdf:Description rdf:about="http://saramago.com">
      <autor:Escreveu  rdf:resource="http://www.books.org/MemorialConvento"/>
      <pessoas:idade>82</pessoas:idade>
   </rdf:Description>
</rdf:RDF>
```

Semantic Web

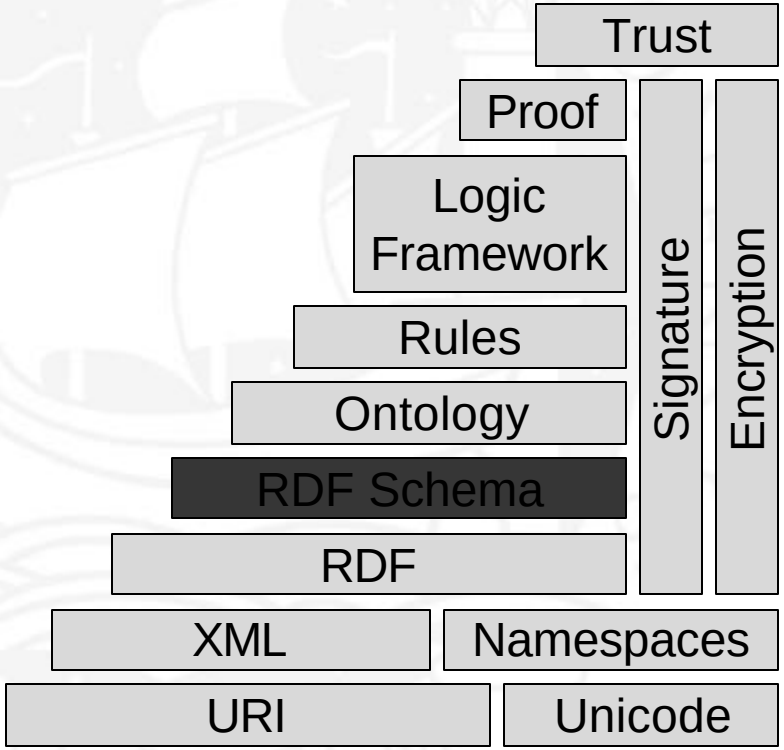
- RDFS – RDF Schema

• Permite definir vocabulários para RDF.

• O RDF permite-nos dizer “qualquer coisa sobre qualquer coisa”, para se obter desambiguidade semântica é preciso definir vocabulários comuns para descrever tipos de coisas.

• RDF Schema permite definir um vocabulário particular para dados RDF (como *escreveu)* e especificar a que tipos de objectos esses atributos podem ser aplicados (como *Autor*).

• Terminologia: Class, subClassOf, Property



Semantic Web

- RDFS – RDF Schema

- Classes, Subclasses e Propriedades

:Animal rdf:type rdfs:Class .

:Dog rdfs:subClassOf :Animal .

:Bobby rdf:type :Dog .

:name rdf:type rdf:Property
:Bobby :name “Bobby” .

:Human rdfs:subClassOf :Animal .
:Zé rdfs:type :Human .

:dono rdf:type rdf:Property .
:Zé :dono :Bobby .

Semantic Web

- RDFS – RDF Schema

• Ranges e Domains

:Book rdf:type rdfs:Class .

:bookTitle rdf:type rdf:Property .

:bookTitle rdfs:domain :Book .

:bookTitle rdfs:range rdfs:Literal .

:Bible rdf:type :Book .

:Bible :bookTitle “A Bíblia” .

A que classe o sujeito de uma tripla que use a propriedade pertence

A que classe o objecto de uma tripla que usa a propriedade pertence

Evitam a promiscuidade das propriedades!

Semantic Web

- DAML+OIL
  - Linguagem criada pelo DARPA como uma linguagem de ontologias e inferência baseada em RDF
  - Permite criar ontologias ainda mais expressivas do que com RDF/S
  - Fornece um modo de dizer coisas como:
    - » Inverso
    - » Propriedades não-ambíguas
    - » Propriedades únicas
    - » Listas
    - » Restricções
    - » Cardinalidades
    - » Disjunções
    - » E mais!

Semantic Web

- DAML+OIL
  - Linguagem criada pelo DARPA como uma linguagem de ontologias e inferência baseada em RDF
  - Permite criar ontologias ainda mais expressivas do que com RDF/S
  - Fornece um modo de dizer coisas como:
    - » Inverso
    - » Propriedades não-ambíguas
    - » Propriedades únicas
    - » Listas
    - » Restricções
    - » Cardinalidades
    - » Disjunções
    - » E mais!

DAML+OIL

- Namespaces

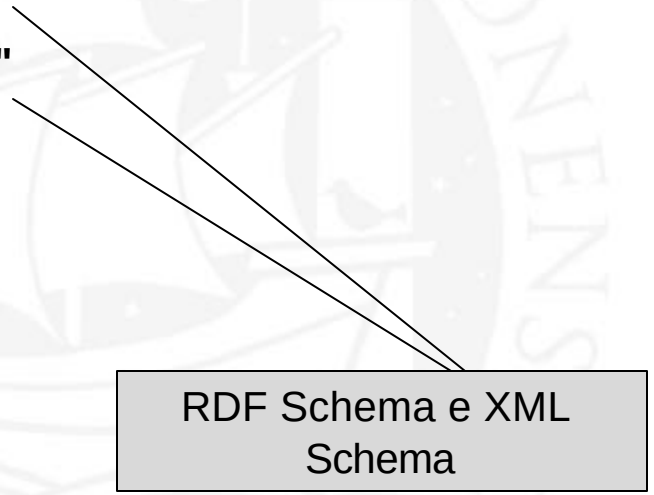
```
<rdf:RDF
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns ="http://www.daml.org/2001/03/daml+oil-ex#"
>
```

Namespace convencional da especificação do RDF

DAML+OIL

- Namespaces

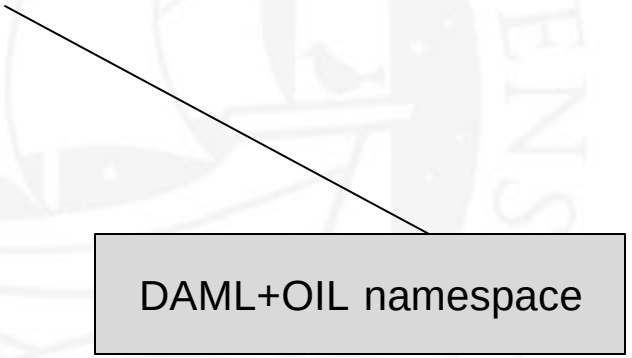
```
<rdf:RDF
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns ="http://www.daml.org/2001/03/daml+oil-ex#"
>
```



DAML+OIL

- Namespaces

```
<rdf:RDF
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns ="http://www.daml.org/2001/03/daml+oil-ex#"
>
```



DAML+OIL

- Namespaces

```
<rdf:RDF
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns ="http://www.daml.org/2001/03/daml+oil-ex#"
>
```

Namespace por defeito.
Refere-se ao próprio documento

DAML+OIL

- Housekeeping

<daml:Ontology rdf:about="">
<daml:versionInfo>
$Id: daml+oil-ex.daml,v 1.9 2001/05/03 16:38:38 mdean Exp $ </daml:versionInfo>
<rdfs:comment>
An example ontology, with data types taken from XML Schema </rdfs:comment>
<daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil"/>
</daml:Ontology>

Isto é uma ontologia!

DAML+OIL

- Housekeeping

```
<daml:Ontology rdf:about="">
<daml:versionInfo>
   $Id: daml+oil-ex.daml,v 1.9 2001/05/03 16:38:38 mdean Exp $ </daml:versionInfo>
<rdfs:comment>
   An example ontology, with data types taken from XML Schema
</rdfs:comment>
<daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil"/>
</daml:Ontology>
```

Duas propriedades para propósito de documentação.

DAML+OIL

- Housekeeping

```
<daml:Ontology rdf:about="">
<daml:versionInfo>
   $Id: daml+oil-ex.daml,v 1.9 2001/05/03 16:38:38 mdean Exp $ </daml:versionInfo>
<rdfs:comment>
   An example ontology, with data types taken from XML Schema
</rdfs:comment>
<daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil"/>
</daml:Ontology>
```

Ontologias importadas. Neste caso é a ontologia DAML+OIL standard.

DAML+OIL

- Definindo classes

Existe uma classe chamada Animal

```
<daml:Class rdf:ID="Animal">
   <rdfs:label>Animal</rdfs:label>
   <rdfs:comment>
      This class of animals is illustrative of a number of ontological idioms.
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Male">
   <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:Class rdf:ID="Female">
   <rdfs:subClassOf rdf:resource="#Animal"/>
   <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Man">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Woman">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Female"/>
</daml:Class>
```

DAML+OIL

- Definindo classes

Damos-lhe um label e um comentário para propósito de documentação.

```
<daml:Class rdf:ID="Animal">
   <rdfs:label>Animal</rdfs:label>
   <rdfs:comment>
      This class of animals is illustrative of a number of ontological idioms.
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Male">
   <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:Class rdf:ID="Female">
   <rdfs:subClassOf rdf:resource="#Animal"/>
   <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Man">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Woman">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Female"/>
</daml:Class>
```

DAML+OIL

- Definindo classes

```
<daml:Class rdf:ID="Animal">
   <rdfs:label>Animal</rdfs:label>
   <rdfs:comment>
      This class of animals is illustrative of a number of ontological idioms.
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Male">
   <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:Class rdf:ID="Female">
   <rdfs:subClassOf rdf:resource="#Animal"/>
   <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Man">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Woman">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Female"/>
</daml:Class>
```

Existe uma classe *Male* que descende de *Animal*.

DAML+OIL

- Definindo classes

```
<daml:Class rdf:ID="Animal">
   <rdfs:label>Animal</rdfs:label>
   <rdfs:comment>
      This class of animals is illustrative of a number of ontological idioms.
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Male">
   <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:Class rdf:ID="Female">
   <rdfs:subClassOf rdf:resource="#Animal"/>
   <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Man">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Woman">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Female"/>
</daml:Class>
```

A classe *Female* também descende de *Animal*, mas é disjunta de *Male*.

DAML+OIL

- Definindo classes

```
<daml:Class rdf:ID="Animal">
   <rdfs:label>Animal</rdfs:label>
   <rdfs:comment>
      This class of animals is illustrative of a number of ontological idioms.
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Male">
   <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:Class rdf:ID="Female">
   <rdfs:subClassOf rdf:resource="#Animal"/>
   <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Man">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Woman">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Female"/>
</daml:Class>
```

Uma classe pode ter várias superclasses.
Man é *Male Person.*

DAML+OIL

- Definindo classes

```
<daml:Class rdf:ID="Animal">
   <rdfs:label>Animal</rdfs:label>
   <rdfs:comment>
      This class of animals is illustrative of a number of ontological idioms.
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Male">
   <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:Class rdf:ID="Female">
   <rdfs:subClassOf rdf:resource="#Animal"/>
   <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Man">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Male"/>
</daml:Class>

<daml:Class rdf:ID="Woman">
   <rdfs:subClassOf rdf:resource="#Person"/>
   <rdfs:subClassOf rdf:resource="#Female"/>
</daml:Class>
```

Uma classe pode ter várias superclasses.
Woman é *Female Person.*

DAML+OIL

- Definindo propriedades (ObjectProperty)

```
<daml:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</daml:ObjectProperty>
```

Relação *hasParent* que vai ser usada para relacionar dois animais.

DAML+OIL

- Definindo propriedades (ObjectProperty)

```
<daml:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</daml:ObjectProperty>
```

Domain define quem vai ser
o sujeito na relação

DAML+OIL

- Definindo propriedades (ObjectProperty)

```
<daml:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</daml:ObjectProperty>
```

Range define quem vai ser o complemento directo na relação.
Animal é pai de Animal.

DAML+OIL

- Definindo propriedades (ObjectProperty)

```
<daml:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</daml:ObjectProperty>
```

hasFather é uma subpropriedade de *hasParent*, (i.e.: quem é *Father* também é *Parent*).

DAML+OIL

- Definindo propriedades (ObjectProperty)

```
<daml:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</daml:ObjectProperty>
```

O *range* restringe o complemento directo apenas a objectos *Male* (i.e.: o *Father* tem que ser *Male)*.

DAML+OIL

- Definindo propriedades (DatatypeProperty)

```
<daml:DatatypeProperty rdf:ID="shoesize">
  <rdfs:comment>
    shoesize is a DatatypeProperty whose range is xsd:decimal.
    shoesize is also a UniqueProperty (can only have one shoesize)
  </rdfs:comment>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#decimal"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="age">
  <rdfs:comment>
    age is a DatatypeProperty whose range is xsd:decimal.
    age is also a UniqueProperty (can only have one age)
  </rdfs:comment>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#nonNegativeInteger"/>
</daml:DatatypeProperty>
```

DatatypeProperty: relacionam propriedades de objectos a valores.

DAML+OIL

- Definindo propriedades (DatatypeProperty)

```
<daml:DatatypeProperty rdf:ID="shoesize">
  <rdfs:comment>
    shoesize is a DatatypeProperty whose range is xsd:decimal.
    shoesize is also a UniqueProperty (can only have one shoesize)
  </rdfs:comment>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#decimal"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="age">
  <rdfs:comment>
    age is a DatatypeProperty whose range is xsd:decimal.
    age is also a UniqueProperty (can only have one age)
  </rdfs:comment>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#nonNegativeInteger"/>
</daml:DatatypeProperty>
```

UniqueProperty: pode haver apenas um *shoesize* por objecto.

DAML+OIL

- Definindo propriedades (DatatypeProperty)

```
<daml:DatatypeProperty rdf:ID="shoesize">
  <rdfs:comment>
    shoesize is a DatatypeProperty whose range is xsd:decimal.
    shoesize is also a UniqueProperty (can only have one shoesize)
  </rdfs:comment>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#decimal"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="age">
  <rdfs:comment>
    age is a DatatypeProperty whose range is xsd:decimal.
    age is also a UniqueProperty (can only have one age)
  </rdfs:comment>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#nonNegativeInteger"/>
</daml:DatatypeProperty>
```

Aceita apenas valores decimais.

DAML+OIL

- Definindo restrições nas propriedades

```
<daml:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasFather"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#shoesize"/>
      <daml:minCardinality>1</daml:minCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Temos uma classe *Person* que descende de *Animal*.

DAML+OIL

- Definindo restrições nas propriedades

```
<daml:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasFather"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#shoesize"/>
      <daml:minCardinality>1</daml:minCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Person só pode ter *Person* em *hasParent*
(i.e.: uma pessoa só pode ter como pai outra pessoa)

DAML+OIL

- Definindo restrições nas propriedades

```
<daml:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasFather"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#shoesize"/>
      <daml:minCardinality>1</daml:minCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Cada pessoa tem que ter exactamente um pai.

DAML+OIL

- Definindo restrições nas propriedades

```
<daml:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasFather"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#shoesize"/>
      <daml:minCardinality>1</daml:minCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Cada pessoa tem que ter pelo menos um tamanho de pé (sintaxe alternativa).

DAML+OIL

- Notações para propriedades

```
<daml:UniqueProperty rdf:ID="hasMother">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Female"/>
</daml:UniqueProperty>

<daml:ObjectProperty rdf:ID="hasChild">
  <daml:inverseOf rdf:resource="#hasParent"/>
</daml:ObjectProperty>

<daml:TransitiveProperty rdf:ID="hasAncestor">
  <rdfs:label>hasAncestor</rdfs:label>
</daml:TransitiveProperty>

<daml:TransitiveProperty rdf:ID="descendant"/>

<daml:ObjectProperty rdf:ID="hasMom">
  <daml:samePropertyAs rdf:resource="#hasMother"/>
</daml:ObjectProperty>
```

UniqueProperty implica cardinalidade 1.

DAML+OIL

- Notações para propriedades

```
<daml:UniqueProperty rdf:ID="hasMother">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Female"/>
</daml:UniqueProperty>

<daml:ObjectProperty rdf:ID="hasChild">
  <daml:inverseOf rdf:resource="#hasParent"/>
</daml:ObjectProperty>

<daml:TransitiveProperty rdf:ID="hasAncestor">
  <rdfs:label>hasAncestor</rdfs:label>
</daml:TransitiveProperty>

<daml:TransitiveProperty rdf:ID="descendant"/>

<daml:ObjectProperty rdf:ID="hasMom">
  <daml:samePropertyAs rdf:resource="#hasMother"/>
</daml:ObjectProperty>
```

inverseOf:
se (*x hasChild y)*, então *(y hasParent x)*

DAML+OIL

- Notações para propriedades

```
<daml:UniqueProperty rdf:ID="hasMother">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Female"/>
</daml:UniqueProperty>

<daml:ObjectProperty rdf:ID="hasChild">
  <daml:inverseOf rdf:resource="#hasParent"/>
</daml:ObjectProperty>

<daml:TransitiveProperty rdf:ID="hasAncestor">
  <rdfs:label>hasAncestor</rdfs:label>
</daml:TransitiveProperty>

<daml:TransitiveProperty rdf:ID="descendant"/>

<daml:ObjectProperty rdf:ID="hasMom">
  <daml:samePropertyAs rdf:resource="#hasMother"/>
</daml:ObjectProperty>
```

TransitiveProperty: declara uma propriedade como sendo transitiva. (nota: aqui falta estabelecer que é transitiva em relação a *hasParent*)

DAML+OIL

- Notações para propriedades

```
<daml:UniqueProperty rdf:ID="hasMother">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Female"/>
</daml:UniqueProperty>

<daml:ObjectProperty rdf:ID="hasChild">
  <daml:inverseOf rdf:resource="#hasParent"/>
</daml:ObjectProperty>

<daml:TransitiveProperty rdf:ID="hasAncestor">
  <rdfs:label>hasAncestor</rdfs:label>
</daml:TransitiveProperty>

<daml:TransitiveProperty rdf:ID="descendant"/>

<daml:ObjectProperty rdf:ID="hasMom">
  <daml:samePropertyAs rdf:resource="#hasMother"/>
</daml:ObjectProperty>
```

samePropertyAs: declara *hasMom* e *hasMother* como sendo sinónimos.

DAML+OIL

- Notações para classes

```
<daml:Disjoint rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Car"/>
    <daml:Class rdf:about="#Person"/>
    <daml:Class rdf:about="#Plant"/>
    <daml:Class rdf:about="#Computer"/>
</daml:Disjoint>
```

```
<daml:Class rdf:about="#Person">
  <rdfs:comment>every person is a man or a woman</rdfs:comment>
  <daml:disjointUnionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Man"/>
    <daml:Class rdf:about="#Woman"/>
  </daml:disjointUnionOf>
</daml:Class>
```

Todas as classes são disjuntas umas das outras.

DAML+OIL

- Notações para classes

```
<daml:Disjoint rdf:parseType=“daml:collection”>
    <daml:Class rdf:about=“#Car”/>
    <daml:Class rdf:about=“#Person”/>
    <daml:Class rdf:about=“#Plant”/>
    <daml:Class rdf:about=“#Computer”/>
</daml:Disjoint>
```

```
<daml:Class rdf:about="#Person">
  <rdfs:comment>every person is a man or a woman</rdfs:comment>
  <daml:disjointUnionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Man"/>
    <daml:Class rdf:about="#Woman"/>
  </daml:disjointUnionOf>
</daml:Class>
```

Identificamos a classe *Person* como sendo a união disjunta de um conjunto de outras classes.

DAML+OIL

- Notações para classes

```
<daml:Class rdf:ID="TallMan">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#TallThing"/>
    <daml:Class rdf:about="#Man"/>
  </daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="MarriedPerson">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Person"/>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasSpouse"/>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="HumanBeing">
  <daml:sameClassAs rdf:resource="#Person"/>
</daml:Class>
```

Podemos construir classes através da intersecção de outras classes.

DAML+OIL

- Notações para classes

```
<daml:Class rdf:ID="TallMan">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#TallThing"/>
    <daml:Class rdf:about="#Man"/>
  </daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="MarriedPerson">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Person"/>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasSpouse"/>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="HumanBeing">
  <daml:sameClassAs rdf:resource="#Person"/>
</daml:Class>
```

Podemos construir classes através da intersecção de classes com propriedades

DAML+OIL

- Notações para classes

```
<daml:Class rdf:ID="TallMan">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#TallThing"/>
    <daml:Class rdf:about="#Man"/>
  </daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="MarriedPerson">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Person"/>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasSpouse"/>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="HumanBeing">
  <daml:sameClassAs rdf:resource="#Person"/>
</daml:Class>
```

Também podemos declarar classes como sendo sinónimas.

DAML+OIL

- Instanciações

```
<Person rdf:ID=“Pinto da costa”>
    <rdfs:label>Pinto da Costa</rdfs:label>
    <rdfs:comment>Pinto da Costa é uma pessoa e um gatuno</rdfs:comment>
    <age:><xsd:integer rdf:value=“66” /></age>
    <shoesize><xsd>integer rdf:value=“42” /></shoesize>
</Person>
```

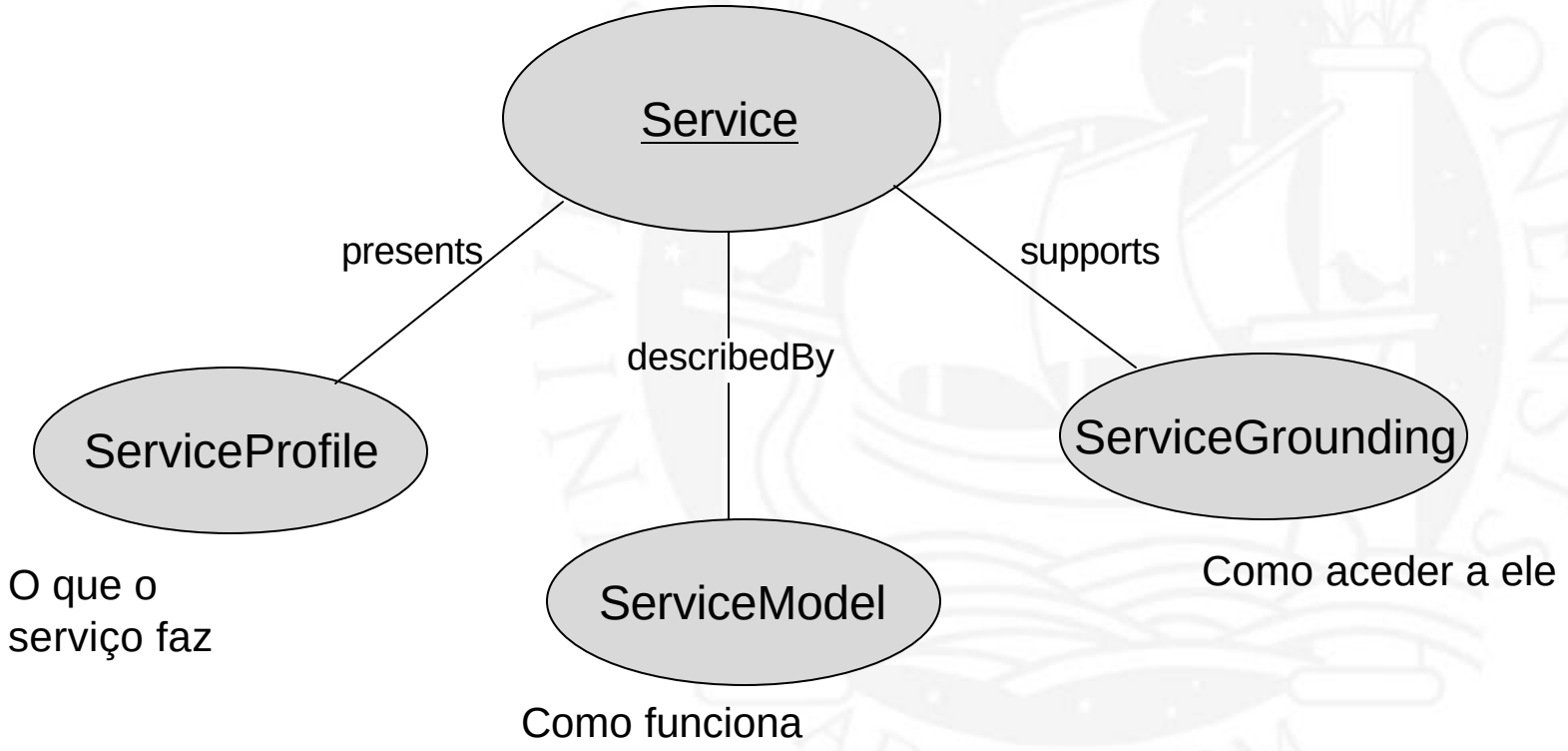
DAML-S

- Ontologias para Web services escritas em DAML+OIL

- Objectivo é automatizarem as seguintes tarefas:
  - Descoberta automática de Web services
  - Invocação automática de Web services
  - Composição automática de Web services
  - Monitorização automática de Web services

DAML-S

- Estrutura da ontologia



DAML-S

- Exemplo trivial
- Congo: um Web service de compra de livros
- Os clientes fornecem informação sobre os livros que pretendem, assim como informação pessoal (morada, CC, etc.)

DAML-S

- Declarando o serviço

```
<service:Service rdf:ID="CongoBuyService">
   <!-- Reference to the Profile -->
   <service:presents rdf:resource="#CongoBuyProfile"/>

   <!-- Reference to the Process Model -->
   <service:describedBy rdf:resource="#CongoBuyProcessModel"/>

   <!-- Reference to the Grounding -->
   <service:supports rdf:resource="#CongoBuyGrounding"/>
</service:Service>
```

Declaração do serviço

DAML-S

- Declarando o serviço

```
<service:Service rdf:ID="CongoBuyService">
   <!-- Reference to the Profile -->
   <service:presents rdf:resource="#CongoBuyProfile"/>

   <!-- Reference to the Process Model -->
   <service:describedBy rdf:resource="#CongoBuyProcessModel"/>

   <!-- Reference to the Grounding -->
   <service:supports rdf:resource="#CongoBuyGrounding"/>
</service:Service>
```

Referências para o *ServiceProfile, ServiceModel* e *ServiceGrounding*.

DAML-S

- Especificando um *ServiceProfile*

<profileHierarchy:BookSelling rdf:ID="CongoBuyProfile">

```
<service:presentedBy rdf:resource="&congoService;#ExpressCongoBuyService"/>

<profile:serviceName>Congo_BookBuying_Agent</profile:serviceName>

<profile:textDescription>
   This agentified service provides the opportunity to browse a
   book selling site and buy books there
</profile:textDescription>
```

Declara o *profile* e enquadra o serviço na taxonomia dos Web services como sendo um serviço de venda de livros.

DAML-S

- Especificando um *ServiceProfile*

```
<profileHierarchy:BookSelling rdf:ID="CongoBuyProfile">

   <service:presentedBy rdf:resource="#CongoBuyService"/>

  <profile:serviceName>Congo_BookBuying_Agent</profile:serviceName>

   <profile:textDescription>
      This agentified service provides the opportunity to browse a
      book selling site and buy books there
   </profile:textDescription>
```

Referência para a classe-pai *Service*

DAML-S

- Especificando um *ServiceProfile*

```
<profileHierarchy:BookSelling rdf:ID=“CongoBuyProfile">

    <service:presentedBy rdf:resource="#CongoBuyService"/>

   <profile:serviceName>Congo Book Selling Service</profile:serviceName>

    <profile:textDescription>
       This agentified service provides the opportunity to buy badass books
    </profile:textDescription>
```

Nome e descrição textual do serviço para leitura humana.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:contactInformation>
    <profile:Actor rdf:ID="CongoBuy_contacts">
      <profile:name>ExpressCongoBuy</profile:name>
      <profile:title> Service Representative </profile:title>
      <profile:phone>412 268 8780 </profile:phone>
      <profile:fax>412 268 5569 </profile:fax>
      <profile:email>Bravo@Bravoair.com</profile:email>
      <profile:physicalAddress>
          somewhere 2,
          OnWeb,
          Montana 52321,
          USA
      </profile:physicalAddress>
      <profile:webURL>
          http://www.daml.org/services/daml-s/0.9/ExpressCongoBuy.html
      </profile:webURL>
    </profile:Actor>
  </profile:contactInformation>
```

Informação de contacto. Descreve o fornecedor do serviço.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:contactInformation>
    <profile:Actor rdf:ID="CongoBuy_contacts">
      <profile:name>ExpressCongoBuy</profile:name>
      <profile:title> Service Representative </profile:title>
      <profile:phone>412 268 8780 </profile:phone>
      <profile:fax>412 268 5569 </profile:fax>
      <profile:email>Bravo@Bravoair.com</profile:email>
      <profile:physicalAddress>
          somewhere 2,
          OnWeb,
          Montana 52321,
          USA
      </profile:physicalAddress>
      <profile:webURL>
          http://www.daml.org/services/daml-s/0.9/ExpressCongoBuy.html
      </profile:webURL>
    </profile:Actor>
  </profile:contactInformation>
```

Todos as entidades são armazenadas cada uma numa classe *Actor*.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:contactInformation>
    <profile:Actor rdf:ID="CongoBuy_contacts">
      <profile:name>ExpressCongoBuy</profile:name>
      <profile:title> Service Representative </profile:title>
      <profile:phone>412 268 8780 </profile:phone>
      <profile:fax>412 268 5569 </profile:fax>
      <profile:email>Bravo@Bravoair.com</profile:email>
      <profile:physicalAddress>
          somewhere 2,
          OnWeb,
          Montana 52321,
          USA
      </profile:physicalAddress>
      <profile:webURL>
          http://www.daml.org/services/daml-s/0.9/ExpressCongoBuy.html
      </profile:webURL>
    </profile:Actor>
  </profile:contactInformation>
```

Nome, título, telefone, fax, email,
endereço físico, web URL.

DAML-S

- Especificando um *ServiceProfile* (cont.)

<profile:geographicRadius rdf:resource=&country#Congo”/>

```
<profile:qualityRating>
   <profile:QualityRating rdf:ID="Congo-Rating">
      <profile:ratingName>SomeRating</profile:ratingName>
      <profile:rating rdf:resource="&concepts;#GoodRating"/>
   </profile:QualityRating>
</profile:qualityRating>
```

Impõe uma limitação à dsitribuição do serviço.
Neste caso só está disponível na Rep.Dem.Congo.

DAML-S

- Especificando um *ServiceProfile* (cont.)

<profile:geographicRadius rdf:resource=&country#Congo”/>

```
<profile:qualityRating>
   <profile:QualityRating rdf:ID="Congo-Rating">
      <profile:ratingName>John’s Rating</profile:ratingName>
      <profile:rating rdf:resource="&concepts;#GoodRating"/>
   </profile:QualityRating>
</profile:qualityRating>
```

Classificação do serviço.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:input>
   <profile:ParameterDescription rdf:ID="ISBN">
      <profile:parameterName> Book's ISBN </profile:parameterName>
      <profile:restrictedTo rdf:resource="&xsd;#integer"/>
      <profile:refersTo rdf:resource="#BuyBookISBN"/>
   </profile:ParameterDescription>
  <profile:ParameterDescription rdf:ID="BuyerInfo">
      <profile:parameterName> Buyer's Information </profile:parameterName>
      <profile:restrictedTo rdf:resource="#Buyer"/>
      <profile:refersTo rdf:resource="#BuyBookISBN"/>
   </profile:ParameterDescription>
</profile:input>
```

Descreve os inputs para o serviço.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:input>
   <profile:ParameterDescription rdf:ID=“ISBN">
      <profile:parameterName> Book’s ISBN </profile:parameterName>
      <profile:restrictedTo rdf:resource="&xsd;#integer"/>
      <profile:refersTo rdf:resource=“#bookISBN"/>
   </profile:ParameterDescription>
  <profile:ParameterDescription rdf:ID=“BuyerInfo">
      <profile:parameterName> Buyer’s Information </profile:parameterName>
      <profile:restrictedTo rdf:resource="#Buyer"/>
      <profile:refersTo rdf:resource=“#bookISBN"/>
   </profile:ParameterDescription>
</profile:input>
```

O primeiro parâmetro é o ISBN do livro. É um número inteiro e refere-se ao processo *BuyBookISBN.*

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:input>
   <profile:ParameterDescription rdf:ID="ISBN">
      <profile:parameterName> Book's ISBN </profile:parameterName>
      <profile:restrictedTo rdf:resource="&xsd;#integer"/>
      <profile:refersTo rdf:resource="#BuyBookISBN"/>
   </profile:ParameterDescription>
  <profile:ParameterDescription rdf:ID="BuyerInfo">
      <profile:parameterName> Buyer's Information </profile:parameterName>
      <profile:restrictedTo rdf:resource="#Buyer"/>
      <profile:refersTo rdf:resource="#BuyBookISBN"/>
   </profile:ParameterDescription>
</profile:input>
```

O segundo parâmetro é a informação do comprador.
É uma instância da classe *Buyer*.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:output>
    <profile:ParameterDescription rdf:ID="EReceipt">
      <profile:parameterName> EReceipt </profile:parameterName>
      <profile:restrictedTo rdf:resource="#EReceipt"/>
      <profile:refersTo rdf:resource="#congoBuyReceipt"/>
    </profile:ParameterDescription>
  </profile:output>
```

Descreve os outputs do serviço.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:output>
    <profile:ParameterDescription rdf:ID="EReceipt">
      <profile:parameterName> EReceipt </profile:parameterName>
      <profile:restrictedTo rdf:resource="#EReceipt"/>
      <profile:refersTo rdf:resource="#congoBuyReceipt"/>
    </profile:ParameterDescription>
  </profile:output>
```

O unico parâmetro é um recibo electrónico que é uma instância da classe *EReceipt*.

DAML-S

- Especificando um *ServiceProfile* (cont.)

```
<profile:output>
    <profile:ParameterDescription rdf:ID="EReceipt">
      <profile:parameterName> EReceipt </profile:parameterName>
      <profile:restrictedTo rdf:resource="#EReceipt"/>
      <profile:refersTo rdf:resource="#congoBuyReceipt"/>
    </profile:ParameterDescription>
  </profile:output>
```

Também poderiam haver precondições. Neste caso uma possível seria o ISBN existir em stock.

```
</profileHierarchy:Bookselling>
```

DAML-S

- Especificando os programas que compõem o serviço (*ServiceModel*)
  - **Existem 3 tipos de processos**
    - **AtomicProcess**
      - É executado em apenas numa chamada (http) que retorna uma resposta (é o nosso caso)
    - **CompositeProcess**
      - Um Web service pode ser composto por vários processos *Atomic* ou *Composite*. São declarados como uma série de construções de controle de fluxo (*sequence, if-then-else, while, etc*).
    - **SimpleProcess**
      - Proporcionam uma abstracção dos processos complexos, permitindo que sejam expostos como uma *Black-box* que esconde os detalhes de *CompositeProcesses*.

DAML-S

- Especificando um processo

```
<daml:Class rdf:ID="BuyBookISBN">
  <rdfs:subClassOf rdf:resource="&process;#AtomicProcess"/>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#ISBN"/>
      <daml:onProperty rdf:resource="#BuyerInfo"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

É um *AtomicProcess*.

DAML-S

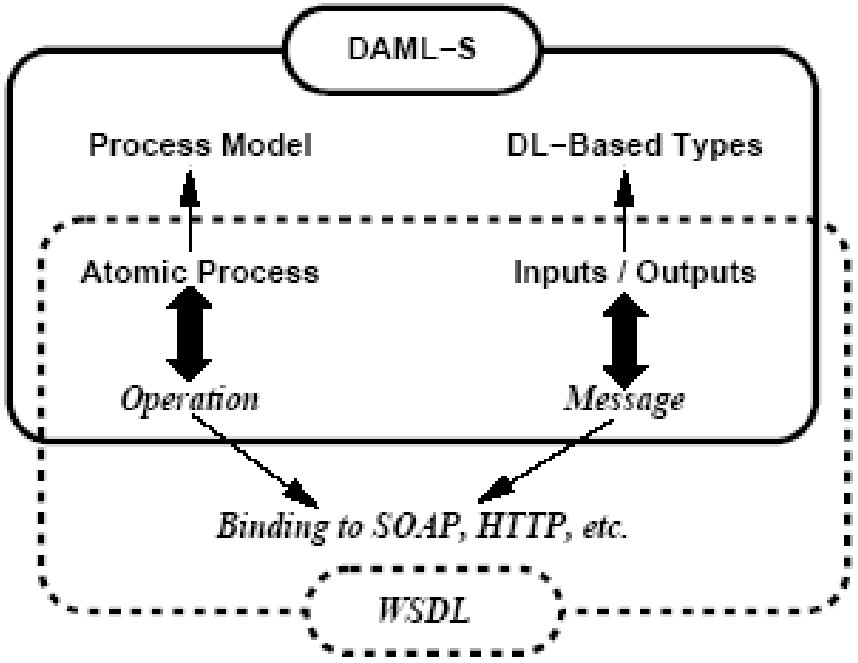
- Especificando um processo

```
<daml:Class rdf:ID="BuyBookISBN">
  <rdfs:subClassOf rdf:resource="&process;#AtomicProcess"/>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#ISBN"/>
      <daml:onProperty rdf:resource="#BuyerInfo"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

A declaração dos inputs/outputs é semelhante à existente no *ServiceProfile*, só que aqui é possível obter maior granularidade especificando *outputs condicionais*, *precondições, efeitos* e outras restrições (cardinalidade, etc).

DAML-S

- *ServiceGrounding*
  - **Efectua a ponte entre o Web service e uma implementação concreta**
  - **Na prática estabelece uma transposição entre o DAML-S e WSDL**
  - **Cada processo atómico vai corresponder a uma operação WSDL**



DAML-S

- *ServiceGrounding*
  - Um processo atómico com inputs e outputs corresponde a uma operação
    WSDL *request-response*.
  - Um processo atómico com inputs, mas sem outputs, corresponde a uma operação WSDL *one-way*.
  - Um processo atómico sem outputs, mas com inputs, corresponde a uma operação WSDL *notification*.
  - Um processo composto com inputs e outpus, mas com o envio dos outputs especificado como acontecendo antes de recepção dos inputs, corresponde a uma operação WSDL *solicit-response* (N/A).

Conclusão

- DAML-S (neste momento substituído pelo OWL) fornece uma ontologia para descrever Web services
- Proporciona aos utilizadores e a agentes de sofware a discoberta, invocação, e composição e monitorização de Web Services
- Começa a atingir um estado relativamente maduro, mas o seu uso está longe de ser vasto
- Muito promissor mas complicado! Principalmente com a nova ontologia OWL.
- Boas ferramentas que permitam construir ontologias de forma simples e rápida serão fundamentais para a expansão da tecnologia.