

# Linguagens Precursoras para Especificação de Ontologias

Prof<sup>a</sup>. Virgínia Brilhante  
DCC/ICE



2005.2

# Formalismos de Representação de Conhecimento

- As linguagens têm como base  
formalismos de representação de conhecimento
  - teorias - fundamentadas em lógica matemática - e sistemas para expressar e manipular conhecimento declarativo de forma tratável e eficiente computacionalmente
  - devem prover:
    - acesso ao conhecimento relevante (descrito, por exemplo, através de fatos e regras)
    - mecanismo(s) de inferência (ou estratégia de resolução)
    - estratégias de controle e escalonamento da inferência

# Formalismos de Representação de Conhecimento

- Dois grandes tipos de formalismos em relação ao foco:
  - **Formalismos orientados a predicados**
    - pioneiros
    - foco em funcionamento
    - ex.: regras, programação em lógica
  - **Formalismos orientados a domínios**
    - foco em expressividade do conhecimento do domínio
    - facilitam a estruturação de conhecimento em termos de classes, relações e restrições
    - ex.: *frames*, redes semânticas, lógicas de descrição

# Linguagens para Especificação de Ontologias

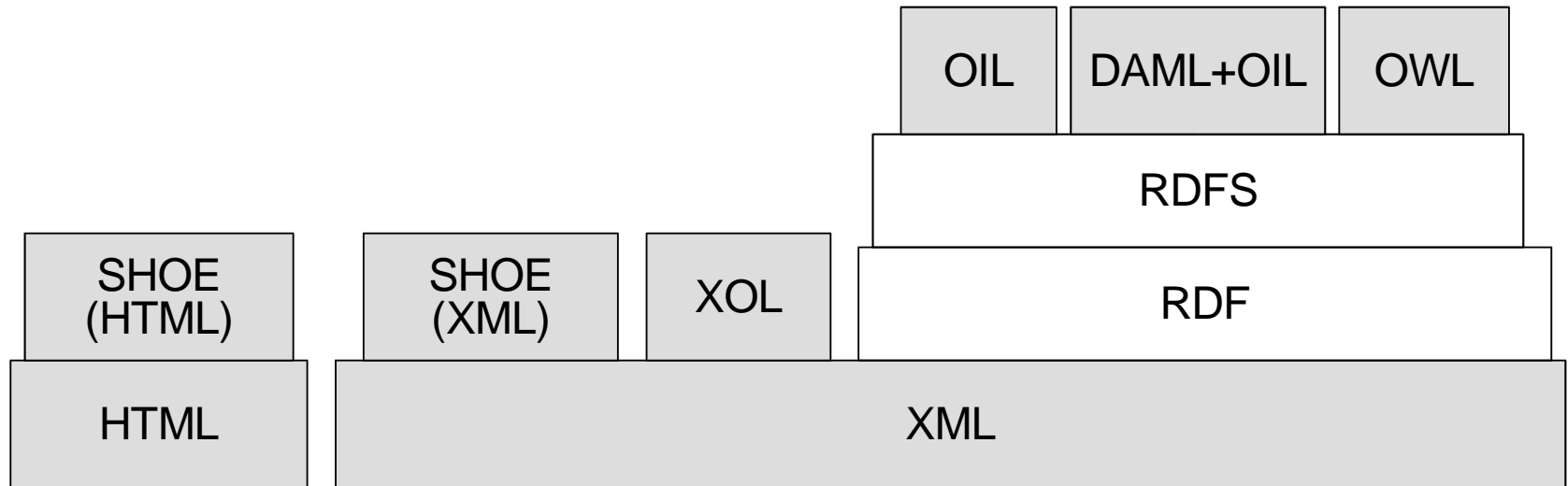
- *Wish list* para as linguagens:
  - expressividade + existência de um motor de inferência
  - acoplamento do motor a um editor de ontologias
  - interoperabilidade com linguagens Web existentes, utilizadas como padrão – ex. XML, RDF
  - traduzível
    - (certa) independência de uma linguagem ou formalismo específico
    - difícil!
  - popularidade

# Linguagens para Especificação de Ontologias

- Existe uma variedade de linguagens
  - algumas com notações gráficas associadas
    - ex. RDF
  - outras não, baseadas em formalismos lógicos
    - **Lógica de Primeira Ordem** (*First Order Logic*)
      - KIF
      - Baseadas em *Frames* combinados com FOL
        - Ontolingua, FLogic
    - **Regras**
      - Prolog (também baseada em FOL), RuleML
    - **Lógicas de Descrição** (*Description Logics*)
      - OIL, DAML+OIL, OWL

# Linguagens para Especificação de Ontologias

- A WWW inspirou a criação das **Ontology Markup Languages**



Fonte: W3C

# XML

- eXtensible Markup Language
- permite a especificação de informações sintáticas (organização, identificação) de dados ou conteúdos de documentos ou recursos web
- independente de h/w, plataforma de s/w ou aplicação
- dados ganham estrutura, mas não significado

```
<library>  
  <book>  
    <title>Being a Dog Is a Full-Time Job</title>  
    <author>Charles M. Schulz</author>  
    <isbn>0836217462</isbn>  
  </book>  
</library>
```

# XML

- Aproxima recursos Web do nível de informação
- Tags (e dados) podem ser acessados por scripts e aplicações em geral, mas...
  - ◆ depende de o programador da aplicação entender o uso das tags pelo marcador do recurso
  - ◆ depende da criação de tags significativas pelo marcador
    - ou seja, o significado dos dados é detido pelo marcador e é, portanto, inacessível para manipulação automatizada



# RDF

- Resource Description Framework
- Linguagem simples para a especificação de significados de conteúdos de um documento ou recurso web (URIs), sem referir-se a estrutura
- Sintaxe semelhante a XML, visando compatibilidade

# RDF

- Modelo de descrição de dados, consiste de:
  - **Recursos**
    - descritos com expressões RDF
    - identificados por URIs
  - **Propriedades** (ou **predicados**)
    - atributos e relações usados para descrever um recurso
  - ***Statements***
    - valores de propriedades de recursos
    - valores podem ser recursos ou literais

# RDF

- Analogia com organização de sentenças em linguagem natural:

sujeito + predicado  
(verbo + objeto)

- ex.: João comprou o ingresso.  
          recurso   propriedade   recurso
- ex.: João disse que Pedro comprou o ingresso. (reificação)
- Algumas primitivas RDF
  - classes: *rdf:Statement*, *rdf:Property*, *rdf:List*
  - propriedades: *rdf:value*, *rdf:type* (define se um recurso é uma classe ou propriedade)

# RDF

- Uma especificação RDF pode ser representada como um grafo direcionado rotulado
  - formalmente, equivalente a uma rede semântica

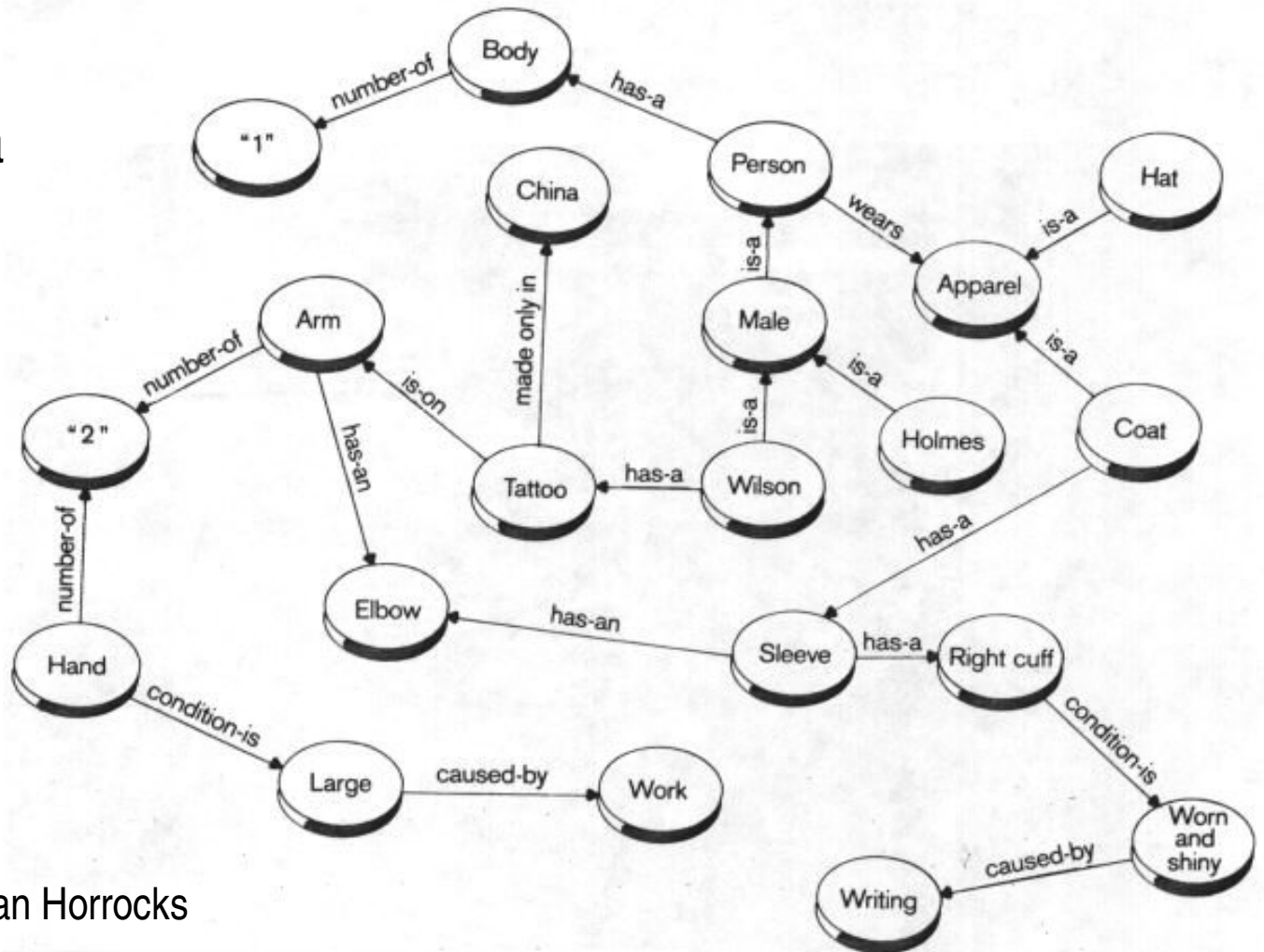
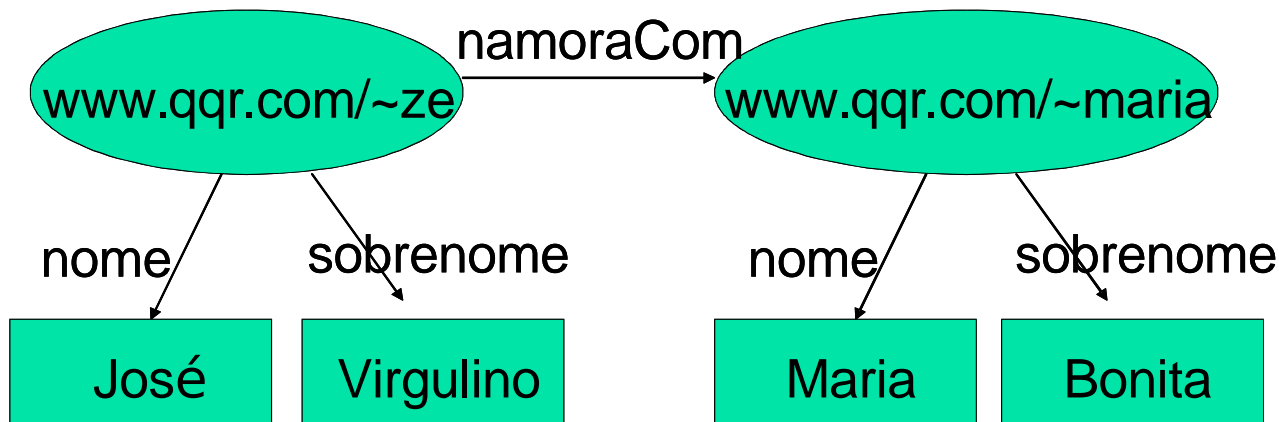


Figura: Ian Horrocks

# RDF



Fonte: Fred Freitas

```
<rdf:Description about=http://www.qqr.com/~ze>
```

```
<nome>Jose</nome>
```

```
<sobrenome>Virgulino</sobrenome>
```

```
<namoraCom>
```

```
<rdf:Description about=http://www.qqr.com/~maria>
```

```
<nome>Maria</nome>
```

```
<sobrenome>Bonita</sobrenome>
```

```
</rdf:Description>
```

```
</namoraCom>
```

```
</rdf:Description>
```

# RDF Schema

- RDF apenas não fornece meta-primitivas
- RDFS criado como uma padronização para o uso de RDF
- Fornece primitivas para a criação de esquemas voltados a aplicações:
  - *rdfs:Class*, *rdfs:Resource* (classes ou conceitos)
  - *rdfs:subClassOf* (propriedade de classe para classe)
  - *rdfs:subPropertyOf*, *rdfs:domain*, *rdfs:range* (outras propriedades)
- A combinação entre RDF e RDF Schema é referenciada como **RDF(S)**

# RDF Schema

- Ex. de código

```
<rdfs:Class rdf:ID="Voo">
```

```
  <rdfs:comment>Uma viagem de aviao</rdfs:comment>
```

```
  <rdfs:subClassOf rdf:resource="#Viagens"/>
```

```
</rdfs:Class>
```

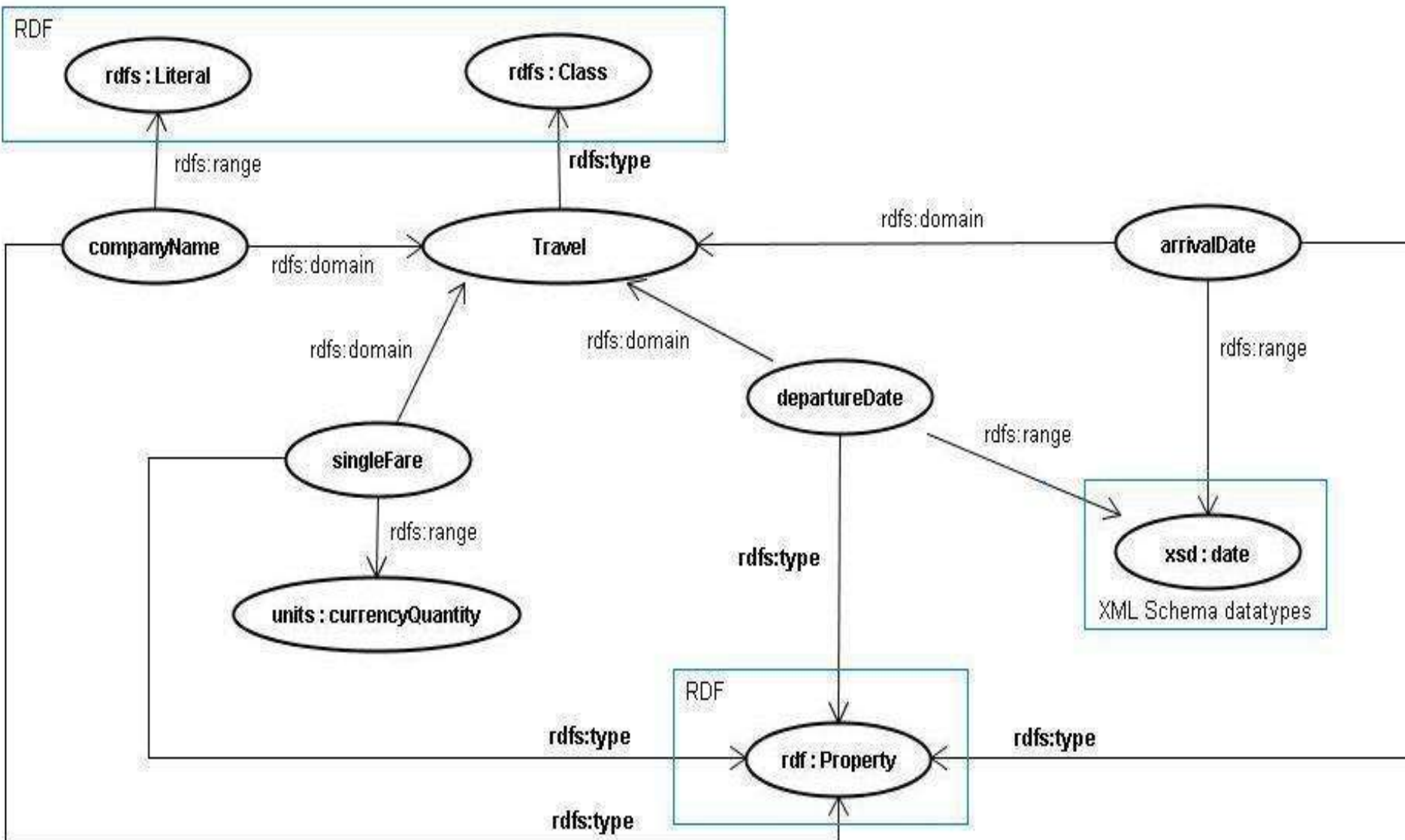
```
<rdf:Property rdf:ID="localDeChegada">
```

```
  <rdfs:domain rdf:resource="#Viagens"/>
```

```
  <rdfs:range rdf:resource="#Local"/>
```

```
</rdf:Property>
```

# RDF/RDF(S)





# RDF/RDF(S)

## Avaliação

- 😊 liberdade para ignorar aspectos estruturais dos recursos
- 😊 são recomendações da W3C e padrões de fato
- 😊 existem ferramentas para edição, validação, consulta, etc.
- 😐 existem mecanismos de inferência, basicamente para consulta
- 😞 ainda não oferecem a expressividade necessária à modelagem de ontologias formais

# KIF

- *Knowledge Interchange Format* [Genesereth e Fikes, 1992]
- Uma linguagem para intercâmbio e representação de conhecimento
- Uma lógica de predicados de primeira ordem pré-fixada
- Pode representar:
  - objetos, permitindo definições de classes e instâncias com especificações de símbolos, números, listas, conjuntos etc.
  - relações e funções n-árias

# KIF

- Exs. de definições

```
(define-class Voo (?X)
```

```
  “Uma viagem de aviao”
```

```
:axiom-def (and (Subclass-Of Voo Viagens)
```

```
  (Template-Facet-Value Cardinalidade
```

```
    numeroVoo Voo 1))
```

```
:class-slots ((numeroVoo Numero)(meioTransporte “aviao”)))
```

```
(define-relation localDeChegada (?viagem ?localChegada)
```

```
  “Uma viagem termina num lugar”
```

```
:def (and (Viagens ?viagem) (Local ?localChegada)))
```

# KIF

- Exs. de definições

**(define-relation** Conexao (?comp1 ?comp2)

“Relacao binária entre dois componentes. Nenhum componente é parte do outro.”

**:def** (**and** (componente ?comp1) (componente ?comp2)

(**not** (parte-de ?comp1 ?comp2) (**not** (parte-de ?comp2 ?comp1))))

**(define-function** Quadrado (?n) --> ?valor

“O quadrado de um número é o seu produto com ele mesmo.”

**:def** (**and** (numero ?n) (numero-nao-negativo ?valor)

**:lambda-body** (\* ?n ?n))

# KIF

- Exs. de definições

**(define-axiom** Unidade-medida-com-expoente

“Uma unidade de medida elevada a um expoente real é também uma unidade de medida.”

**:= (forall ?u ?n (=> (and (unidade-de-medida ?u) (numero-real ?n))  
(unidade-de-medida (expt ?u ?n))))**

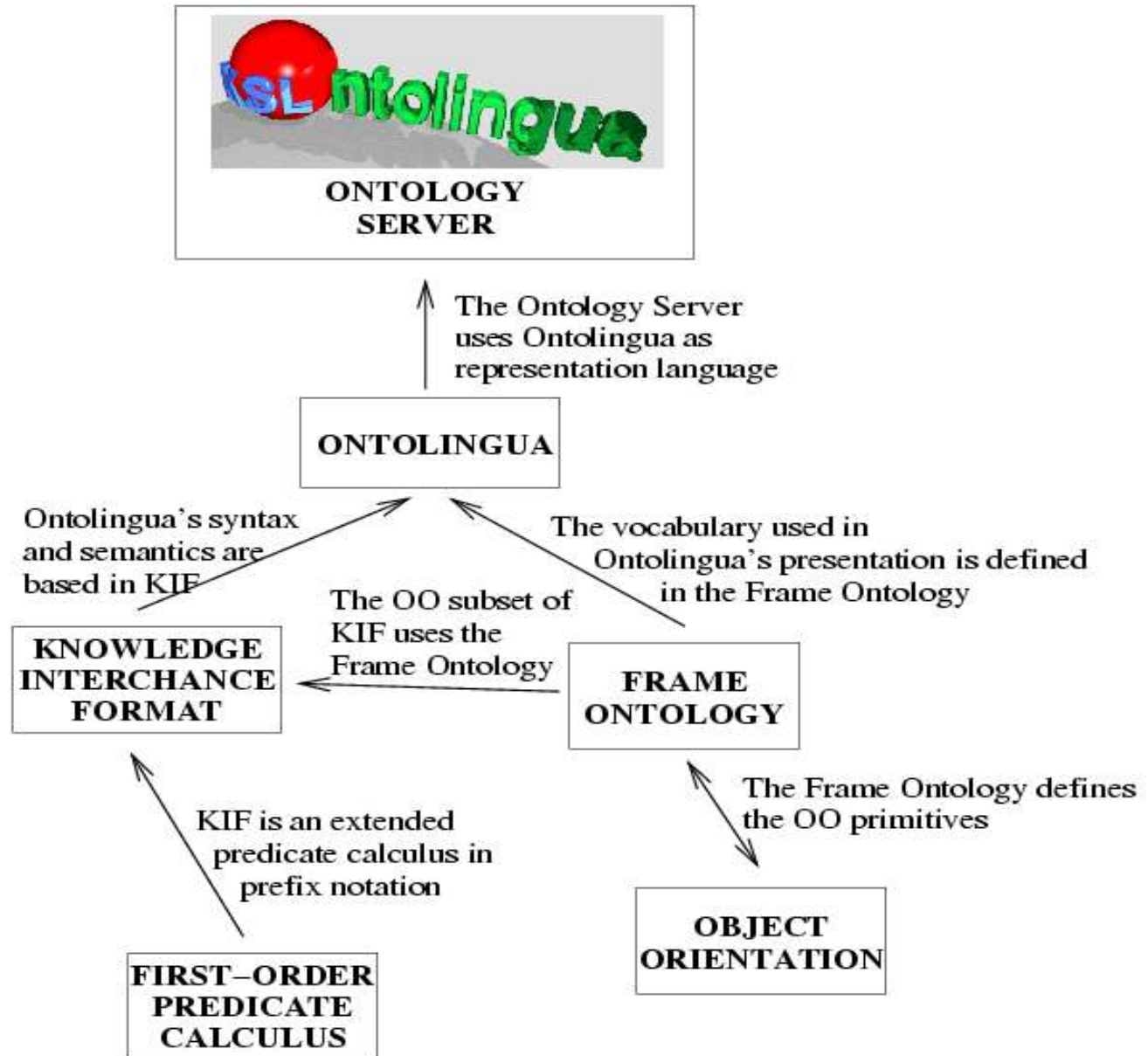
# KIF

- *Knowledge Engineering Effort* (KSE) produziu:
  - KIF, linguagem lógica base para a ...
  - Ontolingua, linguagem para intercâmbio de conhecimento
    - KIF combinada com *frames*
  - Ontolingua Server, ambiente para edição, manipulação e acesso a ontologias --  
<http://www.ksl.stanford.edu/software/ontolingua/>
    - grande biblioteca de ontologias compartilháveis
    - tradutor para linguagens *target*

# KIF

- *Knowledge Engineering Effort* (KSE) produziu: (cont.)
  - **KQML** (*Knowledge Query and Manipulation Language*), linguagem para comunicação entre agentes
  - **OKBC** (*Open Knowledge Base Connectivity*), protocolo para conectividade e interoperabilidade entre ferramentas ou sistemas

# KIF em Ontolingua

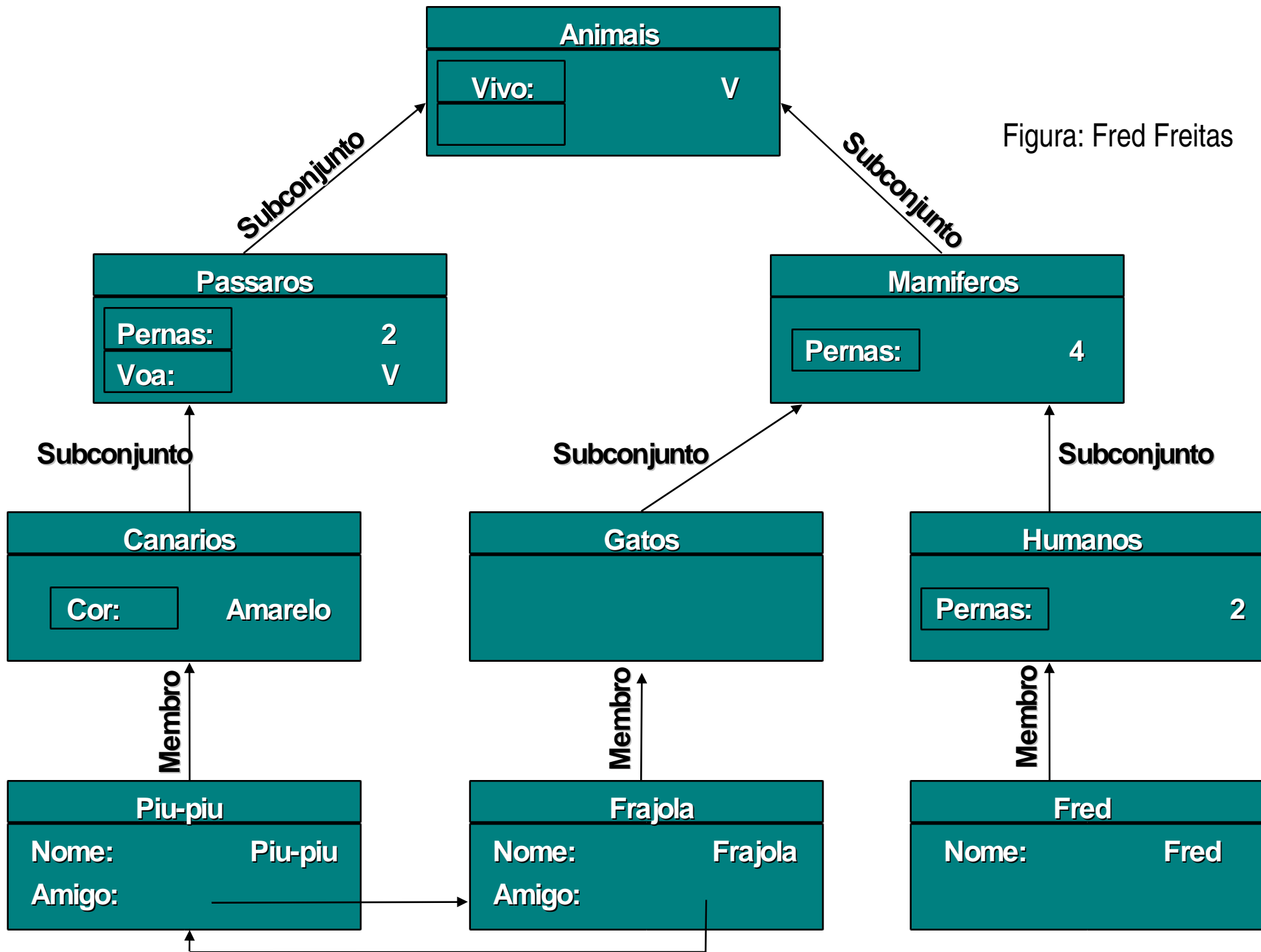




# KIF em Ontologia

- *Frames* (Quadros)
  - Base: modelos mentais de psicologia cognitiva usados na resolução de problemas
    - Esquemas: estruturas de conhecimento (estereótipos) armazenadas na memória duradoura, baseadas em experiências passadas, a serem adaptadas
  - Proposta para representação de conhecimento por M. Minsky em 1975
  - Precursores declarativos das classes no paradigma OO

Figura: Fred Freitas



# KIF em Ontologia

- *Frames* (Quadros) (cont.)
  - Expressividade
    - Classes
      - Herança múltipla
      - Instâncias
    - Atributos (*slots*)
    - Facetas
      - Restrições sobre os *slots*
      - exs. valor default, domínio, cardinalidade, tipo (inteiro, string, booleano, etc.), etc.
  - Inferência por meio de herança e restrições

# KIF em Ontolingua

- Uso de recursos de frame x escrita de axiomas livres

“It is possible for users to write arbitrary axioms in KIF, but it is awkward for them to do so. As a result, most users most of the time write definitions that are readily translated into the object-oriented languages. The principle is: never prevent users from saying what they want to say, but encourage them to say things in a way that is easy to work with.”

[Uschold and Gruninger, 1996]

# KIF em Ontologia

- *Frames* (Quadros) (cont.)
  - Classes (OO) são semelhantes a frames, mas há diferenças
  - Classes e seus objetos:
    - modelam informações do mundo real em estruturas de dados
    - incluem métodos e o conceito de encapsulamento, desnecessários em um paradigma puramente declarativo
    - visam reuso de código
    - incluem detalhes de implementação, como tamanho de strings, etc.

# KIF em Ontolingua

- Problemas da Ontolingua
  - ontologias ficam complexas porque obrigatoriamente referenciam a *Frame-Ontology*
  - motor de inferência JTP agora existe (2003), mas não parece ser largamente utilizado
  - reuso e tradução
    - *pruning* (poda) da ontologia resultante
    - garantia de consistência?
    - diferenças inerentes de expressividade entre os formalismos
      - informações perdidas durante a tradução

# Referências

- Genesereth, M., Fikes, R. (1992) *Knowledge Interchange Format. Version 3.0 Reference Manual*. Report Logic 92-1, Computer Science Department, Stanford University
- Gómez-Pérez, A., Fernández-López, M., Corcho, O. (2004) *Ontological Engineering*. Springer-Verlag, London
- Uschold, M. and Gruninger, Michael (1996) *Ontologies: Principles, Methods and Applications*. The Knowledge Engineering Review 11(2): 93-136