Frames and OWL side by side

Hai WANG
The University of Manchester
Outline

- Introduction
- Major Differences
- Frames or OWL?
- Conclusion
Introduction

Exists two major ontology modeling flavors:

- Frames based formalisms
  - the dominant approach to knowledge modeling
  - e.g. Protege-Frames, Ontolingua

- Description Logics based formalisms
  - Increasingly popular
  - e.g. OWL
Introduction

 Exists two major ontology modeling flavors

 Users confuse them and make mistakes!

  What is the difference?
  What each of them CAN or CANNOT do?
  Which one should I choose?
Introduction

- Exists two major ontology modeling flavors
- Users confuse about them and make mistakes!

Scopes

- DLs flavor -- OWL DL
- Frames flavor -- Protégé Frames
Introduction -- Frames

**Class**
- A class is a set of entities
- A class can be an instance as well

**Slot**
- Describe the properties of classes and instances
- Two ways to be attached to a frame: Template slot and Own slot

**Facet**
- Specify constraints on allowed slot values
Introduction -- OWL

- **Class**
  - Named Class and anonymous classes
  - Class and Individual are disjoint

- **Property**

- **Vocabulary comparison**

<table>
<thead>
<tr>
<th>Concepts in application domains</th>
<th>Frames</th>
<th>OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Class</td>
<td>Class</td>
</tr>
<tr>
<td>Relations</td>
<td>Slot</td>
<td>Property</td>
</tr>
<tr>
<td>Constrain on slot/property values</td>
<td>Facet ≈ Restriction</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Motivation
- Background
- Major Differences
- Frames or OWL?
- Conclusion
**Semantics difference (UNA)**

- Unique Name Assumption: By default, different names refer to different things.

<table>
<thead>
<tr>
<th>Frames</th>
<th>OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

- In OWL, **different** names can refer to the **same** thing.

Matthew
Matt
matthew.horridge
Matthew Horridge
Handsome

Nick
Matty
mhorridge
Example

Two individuals Matthew and Nick are the chefs for the PizzeriaDelDoge, which is an instance of the class PizzaRestaurant.

Frames

PizzeriaDelDoge has EXACTLY TWO chefs.

OWL

PizzeriaDelDoge has AT LEAST ONE chef.

Individual(PizzeriaDelDoge
  type(PizzaRestaurant)
  value(hasChef Matthew)
  value(hasChef Nick))

AllDifferentFrom(Matthew, Nick)
Semantics difference  
(Close World Vs. Open World Reasoning)

Frames
- Adopts CWR
- If a fact is absent from the knowledge base, it is assumed to be false.
- Everything is prohibited until it is permitted.

OWL
- Adopts OWR
- Something is false only if it contradicts other information
- Everything is permitted until it is prohibited.
Semantics difference
(Close World Vs. Open World Reasoning)

Example

Two individuals, Matthew and Nick, are the chefs for the PizzeriaDelDoge, which is an instance of the class Pizza Restaurant.

Frames

PizzeriaDelDoge has EXACTLY TWO chefs.

OWL

PizzeriaDelDoge has AT LEAST TWO chefs.
## Semantics difference (Single model vs. Multiple Models)

### Frames
- **Single Model**
  - For one KB, there exists only *one* model.

- **Multiple Models**
  - For one KB, there could exist *many* models.

<table>
<thead>
<tr>
<th>Single Model</th>
<th>Multiple Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Monotonic</td>
<td>Monotonic</td>
</tr>
<tr>
<td><em>Can't</em> capture incomplete information</td>
<td>Can capture incomplete information</td>
</tr>
<tr>
<td>Less expressive and cannot support negation and disjunction.</td>
<td>More expressive and can support negation and disjunction.</td>
</tr>
</tbody>
</table>

*Comparison between single model and multiple modes*
Implications for Modeling
(Assertion vs. Classification)

Frames

All subclass relations must be asserted explicitly.

OWL

Subclass relations can be inferred based on the class definition.
Implications for Modeling
(Assertion vs. Classification)

Example

VegetarianPizza is any pizza that has only vegetables as its toppings.
MushroomPizza is a pizza with only mushrooms as toppings.
MushroomPizza is a VegetarianPizza.

Frames

(defclass VegetarianPizzaF ....)
(defclass MushroomPizzaF
  (multislot hasToppingF
    (allowed-class MushroomF)
    (IS-A VegetarianPizzaF)))

VegetarianPizzaF has only necessary definition.
Assert explicitly that MushroomPizza is a subclass of VegetarianPizza

OWL

Class (VegetarianPizza COMPLETE Pizza
  (restriction hasTopping allValuesFrom Vegetable))
Class (MushroomPizza partial Pizza
  restriction (hasTopping allValuesFrom Mushroom))

⇒ subclassof
  (MushroomPizza, VegetarianPizza)

VegetarianPizza has a sufficient definition.
The subclass relation will be inferred.
## Implications for Modeling
*(Constraint vs. Consistency checking)*

<table>
<thead>
<tr>
<th>Frames</th>
<th>OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint checking</td>
<td>Consistency checking.</td>
</tr>
<tr>
<td>Check whether slot values for instances of a class is <strong>valid</strong>.</td>
<td><strong>All</strong> the asserted axioms are valid.</td>
</tr>
<tr>
<td></td>
<td>Check if there is a model that satisfies all the assertions.</td>
</tr>
</tbody>
</table>

**Major statements playing different roles:**
- Facets and property restrictions;
- Domains and ranges of slots and properties;
Implications for Modeling
(Assertion vs. Consistency checking)

Example

The slot/property **hasTopping** has the domain as Pizza.
Choc ice-cream has toppings.

**Frames**

\[
\text{(multislot hasTopping}^F \\
\text{(Type instance)} \\
\text{(domain Pizza}^F )) \\
\text{(defclass ChocIcecream}^F \\
\text{(multislot hasTopping}^F ....) \times
\]

It is an **ERROR** in Frames!

**OWL**

\[
\text{ObjectProperty (hasTopping domain Pizza)}
\]

\[
\text{Class (ChocIcecream partial restriction (hasTopping someValuesFrom Chocolate))} \\
\Rightarrow \text{subclassof(ChocIcecream, Pizza)}
\]

All the asserted axioms are assumed to be right. Infer that ChocIcecream is a subclass of Pizza.
Frames
Two Steps to add constrains to classes:
1. add the slot to the class.
2. associate a facet to the template slot

OWL
Restriction can be associated with a class directly.
Expressiveness Power
(Frames)

- Meta-modeling
- Classes as property values
- Default information and exception
Expressiveness Power (OWL)

- Defined classes
- Embedding class Definition (anonymous classes)
- Set combination on classes
- Characters of Properties
  - Functional, symmetric (Allowed in Frames)
  - Transitive (Not allowed in Frames)
  - OWL 1.1: reflexive, irreflexive, symmetric, and anti-symmetric
Outline

- Motivation
- Background
- Major Differences
- Frames or OWL?
- Conclusion
Frames or OWL? -- Some Guidelines

Frames
An application where:
- The closed-world assumption is appropriate.
- Focuses on data acquisition on instances.
- Requires constraints on slot values.
- Meta-modeling is important.

OWL
An application where:
- The open-world assumption is appropriate.
- New classes have been built from the combinations of other classes.
- Logical consistency needs to be ensured.
- Published on the Semantic Web and accessed by other applications.
- Complicated class hierarchy need to be maintained.
Conclusion

- Motivation
- Background
- Similarities and Differences
- Conclusion Frames or OWL?

Conclusion
Conclusion

- Semantic difference
  - UNA
  - Closed world vs Open World Assumption
  - Single Vs. Multiple models
- Implication
  - Assertion vs Classification
  - Property association
  - Constrain checking vs Reasoning
- Expressive Power
- Others ...
Thank You.