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

	Frames	OWL
Concepts in application domains	Class	Class
Relations	Slot	Property
Constrain on slot/property values	Facet	≈ Restriction

Outline

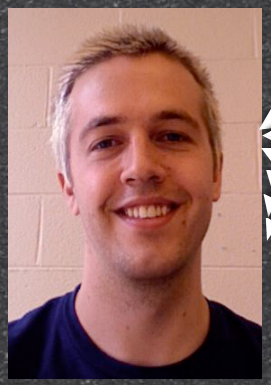
- Motivation
- Background
- **Major Differences**
- Frames or OWL?
- Conclusion

Semantics difference (UNA)

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

Frames	OWL
YES	NO

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



Matthew

Nick

Matt

matthew.horridge

Matty

Matthew Horridge

mhorridge

Handsome

Semantics difference (UNA)

- Example

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

Frames

...
([PizzeriaDelDoge^F] of PizzaRestaurant^F
(hasChef^F [Matthew^F]
[Nick^F])

PizzeriaDelDoge has EXACTLY TWO chefs.

OWL

...
Individual(PizzeriaDelDoge
type(PizzaRestaurant)
value(hasChef Matthew)
value(hasChef Nick))
*AllDifferentFrom(*Matthew, Nick)

PizzeriaDelDoge has AT LEAST ONE chefs.

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^F] of PizzaRestaurant^F
(hasChef^F [Nick^F]
[Matthew^F])

PizzeriaDelDoge has EXACTLY TWO chefs.

OWL

...
Individual(PizzeriaDelDoge
type(PizzaRestaurant)
value(hasChef Matthew)
value(hasChef Nick)
AllDifferentFrom(Matthew, Nick))
Type(restriction (
hasChef
allValuesFrom(oneof(Matthew, Nick))))

PizzeriaDelDoge has AT LEAST TWO chefs.

Semantics difference
(Single model vs. Multiple Models)

Frames

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

OWL

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

Single Model	Multiple Models
Non-Monotonic	Monotonic
Can't capture incomplete information	Can capture incomplete information
Less expressive and cannot support negation and disjunction.	More expressive and can support negation and disjunction.

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* VegetarianPizza^F ....)

(defclass MushroomPizza^F
(multislot hasTopping^F
(allowed-class Mushroom^F)
(IS-A VegetarianPizza^F))

VegetarianPizza^F 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)

Frames

- Constraint checking
- Check whether slot values for instances of a class is valid.

OWL

- Consistency checking.
- All the asserted axioms are valid
- Check if there is a model that satisfies all the assertions.

- 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

(*multislot* hasTopping^F
(*Type* instance)
(domain Pizza^F))
(defclass ChocIcecream^F
(multislot hasTopping^F ....) ✘

It is an ERROR in Frames!

OWL

ObjectProperty (hasTopping domain Pizza)

Class (ChocIcecream partial
restriction (hasTopping someValuesFrom
Chocolate))
⇒ subClassof(ChocIcecream, Pizza)

All the asserted axioms are assumed to be right. Infer that ChocIcecream is a subclass of Pizza.

Implications for Modeling

(Associate of properties/slots)

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.