

# Working with Multiple Ontologies on the Semantic Web

Bernardo Cuenca Grau, Bijan Parsia,  
Evren Sirin

*MINDSWAP Research Group*

*University of Maryland, College Park*

# URIs in OWL Ontologies

- Three key modalities of URI use
  - URIs as data (mention)
  - URIs as identifiers (use)
  - URIs as values of owl:imports (use++)
- owl:imports supports *transclusion*
  - The transclusion is *flat*
    - I.e., an include; imported axioms just asserted

## owl:imports

- Problems with owl:imports
  - Does not support information hiding or filtering
  - None of the imported axioms retain their context

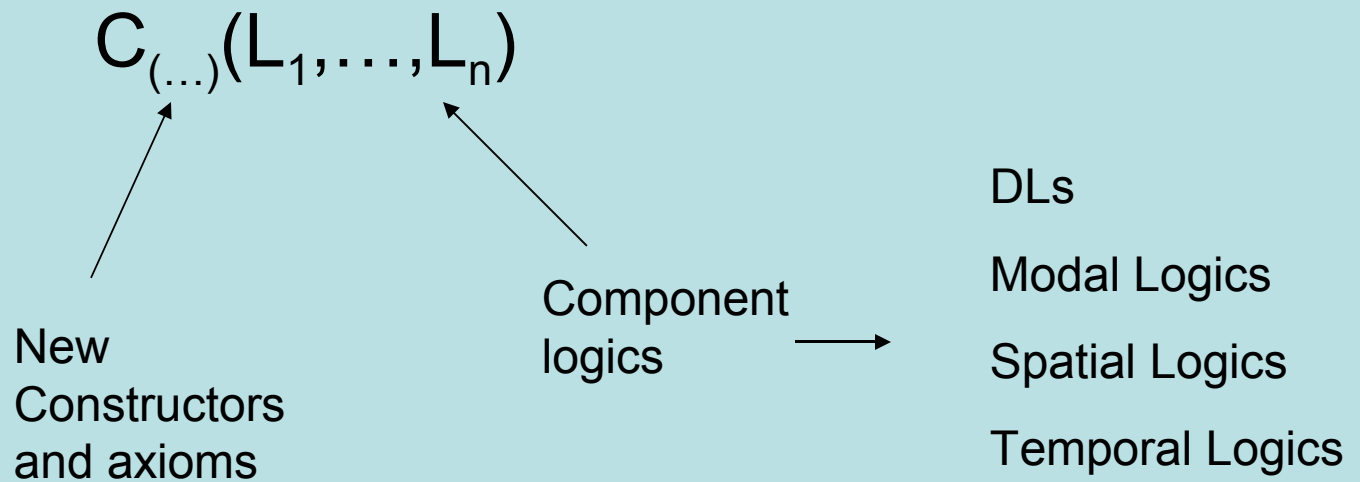
It gives us either ALL or NOTHING

## owl:imports(2)

- Without inclusion, *nothing* from the URI owner (or anyone else) gets in
  - The only sharing is either out of OWL's scope, or conditional. i.e., *if* you import or merge, then the URIs merge as well.
- With inclusion, *everything* gets in
  - Even things which are intuitively irrelevant
  - The resulting ontology's logic is at least as complex as the most complex of any of the included (and could be worse)
  - The resulting ontology *itself* is very complex

# E-Connections: The Basics

- An E-Connection is a knowledge representation language defined as a combination of other logical formalisms.



## E-connections: Combined KBs

- A Combined KB is a set of ontologies written in the language of an E-Connection
- Each component ontology can be written in any of the component logics
- Each component ontology is interpreted in a different logical context

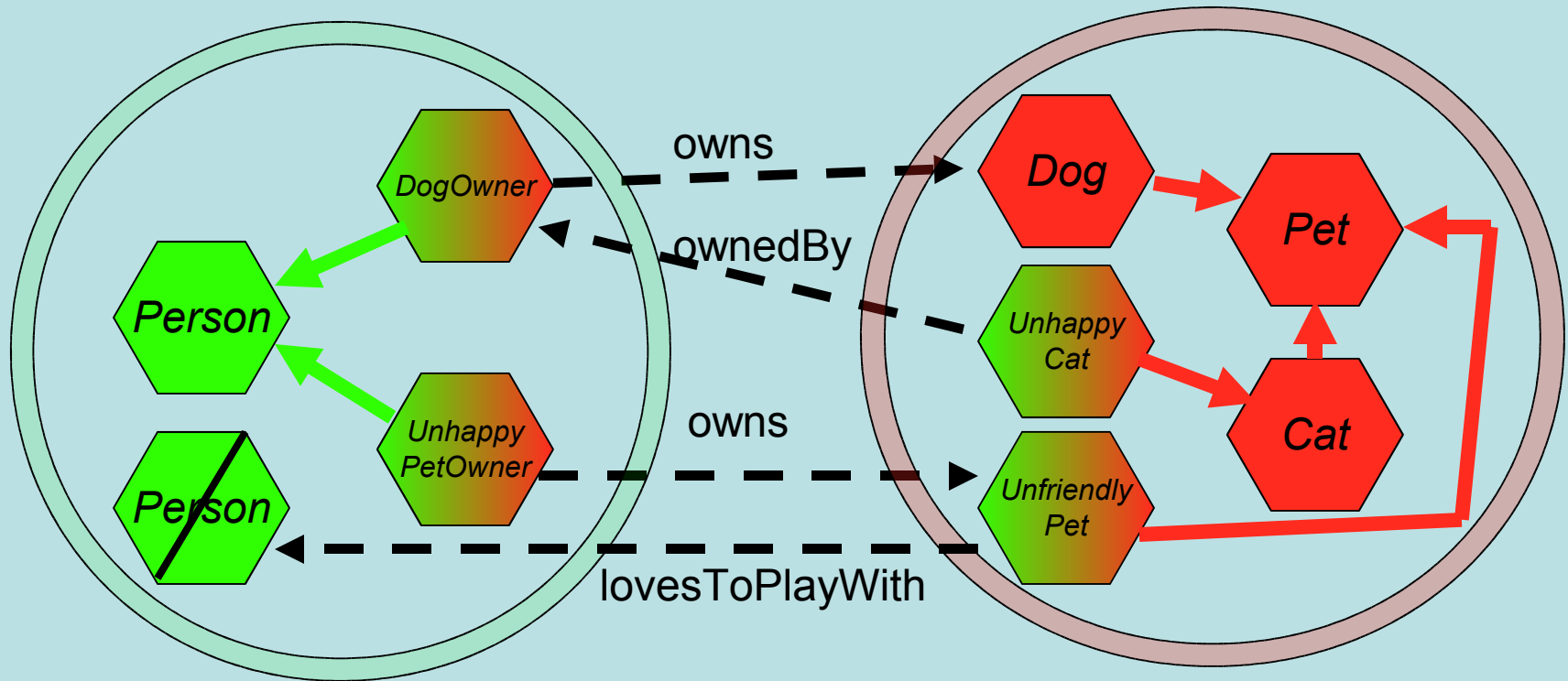
## E-connections: Connected KBs

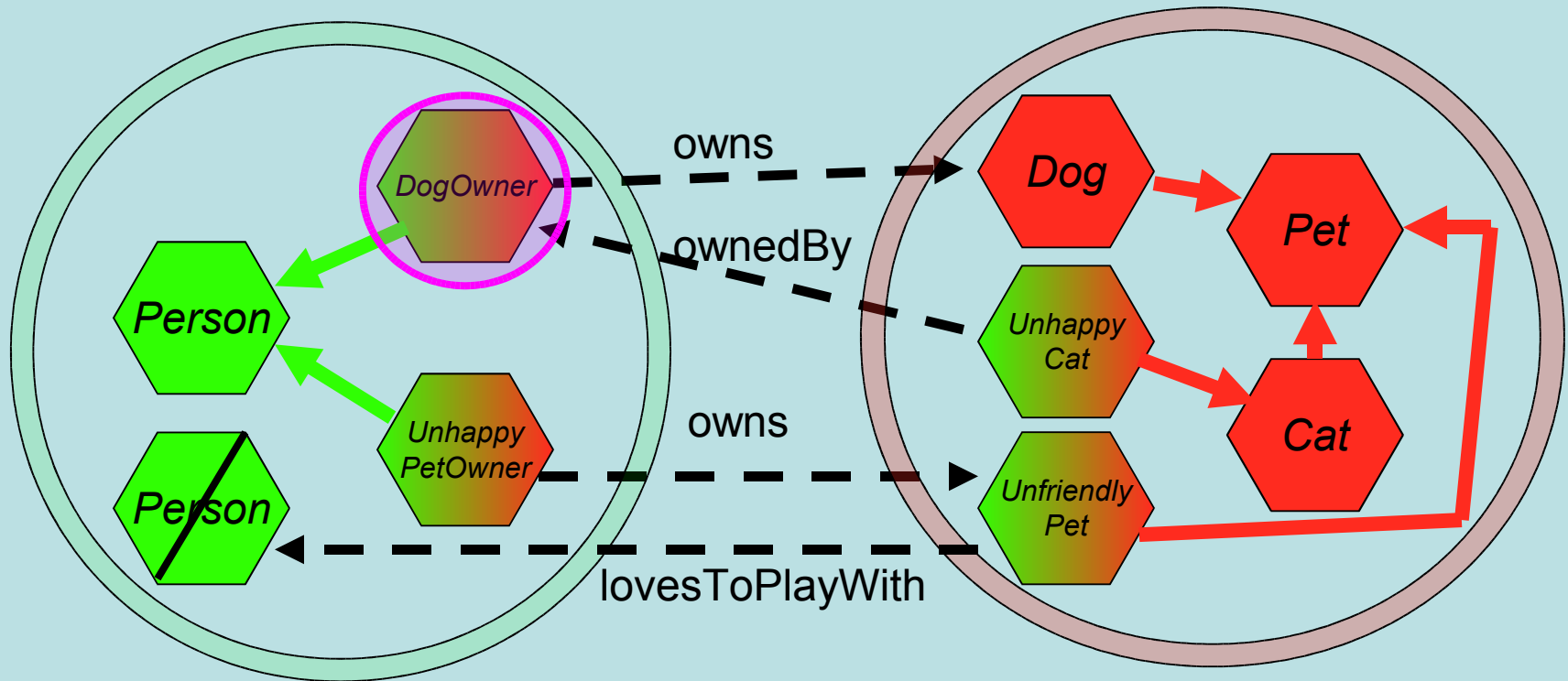
- Logical contexts (domains) are disjoint, but related through “links”
  - Individuals in the source are linked to individuals in the target
- Concepts in each component can be defined in terms of the links
  - In addition to the constructors of the component formalism, there is a set of constructors on the links

# What can we gain?

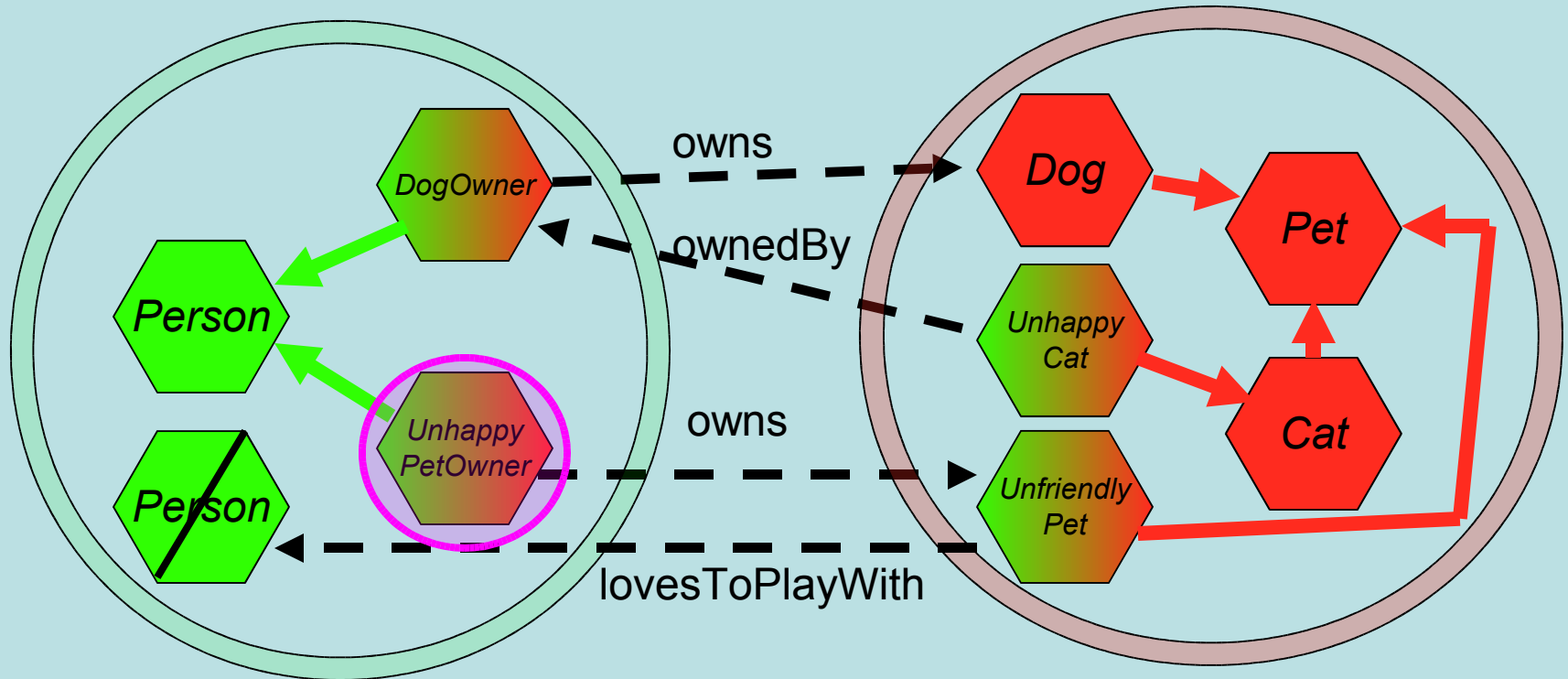
- Expressivity & Decidability
  - Description Logic with temporal or spatial logic
- Expressivity & practical algorithms
  - ex: C(SHIQ,SHOQ,SHIO) merges to SHOIQ
- Modularity: Even in C(SHIF)!
  - Ability to integrate ontologies as reusable modules
  - Ability to split up large ontologies

## Integration: People and Pets example

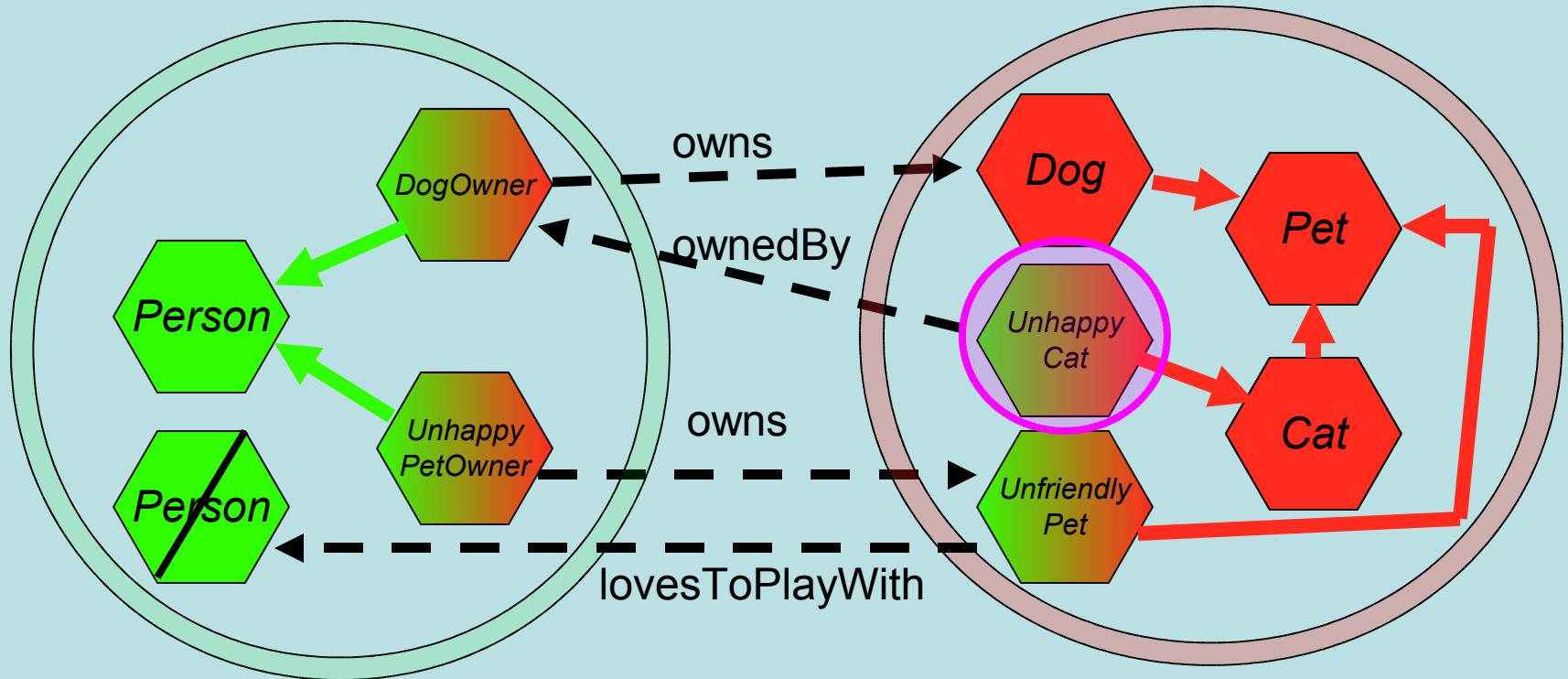




- $\text{DogOwner} = \text{Person} \sqcap \exists \text{owns.Dog}$
- (“owns” is a *link*)



UnhappyPetOwner = Person  $\sqcap$   
 $\exists$ owns.(UnfriendlyPet)



UnhappyCat = Cat  $\sqcap$

$\exists \text{ownedBy} . (\text{DogOwner})$

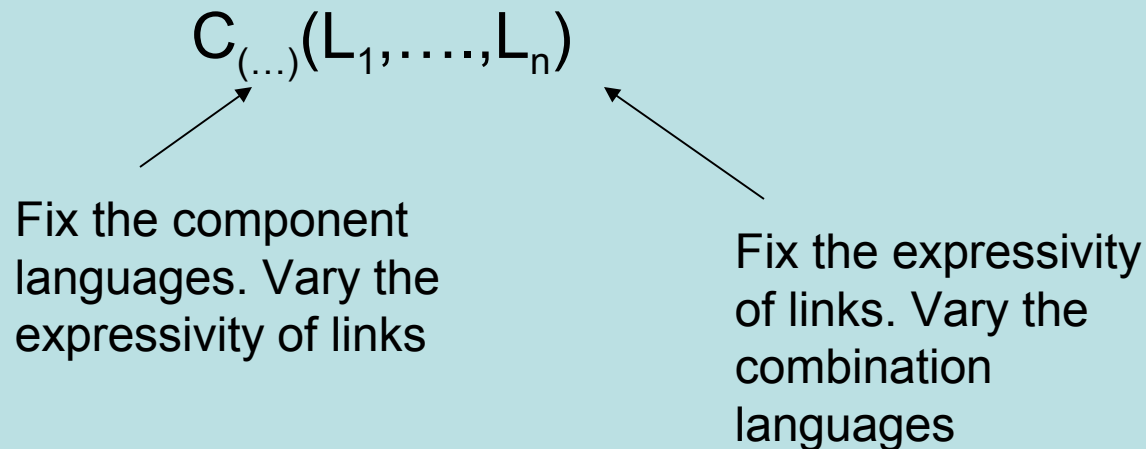
$\text{ownedBy} \text{ inverseOf}(\text{owns})$

# Factoring ontologies

- Ontologies with core and many side lines
  - National Cancer Institute Ontology
    - Char-grilled and belief systems
  - Wine ontology
    - Wines, regions, colors, etc.
- Refactor these
  - Smaller, linked ontologies
  - Each ontology is more focused
  - Easier to understand, evolve, and reuse
  - Possible performance gain

# Families of E-Connection Languages

- Two ways of defining new combination languages:



# PECs...The New ALC

- ALC -- The fundamental v. expressive DL
  - Closed under negation
  - Contains the fundamentals:  $\&$ ,  $\vee$ ,  $\sim$ ,  $\forall$ ,  $\exists$
  - Easy to extend
- Perspectival E-connections  $C(L_1, \dots, L_n)$ : The fundamental E-Connection Language:
  - Existential and value restrictions on links
  - Extensible to more expressive languages

## Extensions of PECs

- Basic E-Connections:  $C_i(L_1, \dots, L_n)$ 
  - Add Inverses on Links
- Extensions with number restrictions
  - FanaticPetOwner = PetOwner  $\sqcap \geq_{20}$  owns.Pet
- Extensions with Link hierarchies
  - lovesToPlayWith  $\sqsubseteq$  lovesActivity
- Extensions with Booleans on Links
  - FrustratedPetOwner = PetOwner  $\sqcap$   
 $\exists(\text{owns} \sqcap \neg \text{likes}).\text{Pet}$

## Reasoning with Econnections of DLs

- Depends on:
  - Which are the component logics
  - Expressivity of the links

If the component logics do not contain nominals, a “black box” technique can be used

## Algorithms

- Translate into background DL
  - But lose many of the advantages
- Direct tableau algorithm
  - Straightforward extension
    - Color the nodes
    - Apply standard techniques to source, links, and target “separately”
  - Doesn’t perturb optimizations (it seems)
- Actually implemented quickly
  - 400 lines with comments in Pellet
  - Helper functions a big chunk

# Benefits!

- More expressivity
  - Larger subset of SHION(D)/OWL DL
    - i.e.,  $C(\text{SHIN}, \text{SHON}, \text{SHIO})$
    - But (now) known sound and complete algorithm
- Pragmatically helpful
  - Links feel semwebby
  - Factored ontologies easier to work with
  - Automatic disjointness
- Performance gains possible
  - Partition nominals, individuals, and CGIs
  - Push expressivity into the links

# Integrating E-Connections in OWL

```
<owl:Class rdf:ID="PetOwner">
  <rdfs:comment>A Person who owns at least one pet</rdfs:comment>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty>
        <owl:LinkProperty rdf:about="#owns">
          <owl:foreignOntology rdf:resource="&pets;"/>
        </owl:LinkProperty>
      <owl:onProperty>
        <owl:someValuesFrom>
          <owl:ForeignClass rdf:about="&pets;#Pet">
            <owl:foreignOntology rdf:resource="&pets;"/>
          </owl:ForeignClass>
        </owl:someValuesFrom>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
```

# Expressivity issues

- Links cannot be transitive
  - Some refactoring blocked
    - But there are often workarounds
  - Cannot be used to simulate subsumption
- Links indexed by source and target
  - So owns to pets and owns to furniture *not* the same!
- Non partitionable domains
  - Suppose we add “Animal” to people and pets?
- Number restrictions with inverses?

## Future Work

- The work described here has been (largely) completed
- Modeling experience
- Empirical evaluation
- Optimizations
- Extend to rules?
  - Datalog isn't an ADS
    - But some prior work is suggestive

## Play with it!

- Web form access to E-conn savvy Pellet
  - <http://www.mindswap.org/2003/pellet/demo>
- Example of refactored ontologies
  - <http://www.mindswap.org/2004/multipleOnt/Fac>
- Swoop support coming soon!
  - Not just editing, but refactoring assistance
  - <http://www.mindswap.org/2004/SWOOP/>