What to Do with an OWL Reasoner:
Introduction to Pellet

Evren Sirin
Clark & Parsia, LLC
What is Pellet?

• Pellet is an OWL-DL reasoner
  ○ Covers all constructs in OWL-DL
  ○ Supports nearly all of OWL 2
• Written in Java and available as open-source
  ○ http://pellet.owldl.com/
• Bindings to popular toolkits and editors
  ○ Jena, OWL-API, Protege, TopBraid Composer
• Next release will be dual licensed (think MySQL)
Pellet History

• Started around 2003 at MINDSWAP group of Univ of Maryland, College Park
  ◦ Initiated by Bijan Parsia
  ◦ Implemented by Evren Sirin
• As of late 2006 development shifted to C&P
  ◦ Commercial support provided
  ◦ Contributions and support from: NCI, NATO NC3A, Lockheed Martin, Clark & Parsia LLC
Talk Outline

• Reasoning in OWL
  ◦ Basic introduction
• Introduction to Pellet
  ◦ Standard features and extensions
• Pellet demo
  ◦ See it in action!
• Example applications
  ◦ Based on first-hand experience
Reasoning in OWL

• Two main reasoning services in OWL
• Given a set of asserted statements
  ○ Check their *consistency*
  ○ *Infer* new conclusions
Simple Inference Example

• From statements
  ○ Penguin subClassOf Bird
  ○ Pablo type Penguin

• Infer that
  ○ Pablo type Bird
(In)consistency Example

• Statements
  - Bird subClassOf FlyingThing
  - Penguin subClassOf Bird
  - Penguin disjointWith FlyingThing

• are inconsistent because penguins are inferred to be both flying and not flying
Complexity

- Things get complicated very quickly
- Many axioms and constructors
  - subClassOf, equivalentClass, subPropertyOf, domain, range, ...
  - and, or, not, some, all, min, max, ...
  - functional, transitive, symmetric, ...
- Algorithmic complexity are high
  - Reasoning algorithms are NEXPTIME
  - Many sophisticated optimizations
Not-so-simple example (1)

- VX_Precursor subClassOf NerveAgentPrecursor
- NerveAgentPrecursor subClassOf Precursor
- Precursor subClassOf ChemicalElement
- ChemicalElement disjoint PublishedWork
- VR_RelatedPublishedWork subClassOf NerveAgentRelatedPublishedWork
- VR_RelatedPublishedWork equivalentClass (refersToPrecursor only VR_Precursor)
- refersToPrecursor subPropertyOf refersTo
- refersTo domain PublishedWork
- NerveAgentRelatedPublishedWork subClassOf PublishedWork
Not-so-simple example (2)

- **OceanCrustLayer** subClassOf **CrustLayer**
- **CrustLayer** subClassOf **LithosphereLayer**
- **LithosphereLayer** subClassOf **SolidEarthLayer**
- **SolidEarthLayer** subClassOf **Layer**
- **Layer** subClassOf **GeometricalObject_3D**
- **GeometricalObject_3D** subClassOf (**hasDimension** value "3")
- **Functional** (**hasDimension**) **OceanCrustLayer** subClassOf **OceanRegion**
- **OceanRegion** subClassOf **TopographicalRegion**
- **TopographicalRegion** subClassOf **EarthRegion**
- **EarthRegion** subClassOf **Region**
- **Region** subClassOf **GeometricalObject_2D**
- **GeometricalObject_2D** subClassOf (**hasDimension** value "2")
Pellet Features

- Standard reasoning features
  - Consistency, classification, realization
- Conjunctive query answering
  - SPARQL-DL queries
- Datatype reasoning
  - User-defined datatypes
  - N-ary datatype predicates
- Rules support
  - DL-safe SWRL rules
- Explanation and debugging
  - Axiom pinpointing service
SPARQL-DL

- Extension of SPARQL query language with OWL-DL entailment regime
  - Defines some restrictions on SPARQL
  - Introduces some new keywords for querying
- Expressive conjunctive query language
- Example: Find all students who are also employees and find out who they work for

```sparql
SELECT * WHERE {
    ?x rdf:type :Student ;
    rdf:type ?C;
    :worksFor ?organization .
    ?C sparqldl:properSubClassOf :Employee . }
```
Datatype Reasoning

• Define classes by restrictions on datatypes
• Comparison can be...
  ○ ... on a single property value
    ▪ A small monitor has a screen size <= 15in
    ▪ Free shipping available for monitors with screen size <= 21in
  ○ ... a combination of properties (OWL 2)
    ▪ A widescreen monitor has height width ratio < .75
• Based on datatype restrictions reasoner can infer
  ○ type relations: 14in monitor is a small monitor
  ○ subclass relations: small monitors ship free
SWRL Rules

• Extends expressivity of OWL
  ○ Comes with a price: undecidability
• Pellet supports DL-safe rules
  ○ Rules apply to only instances
• Example rule
  ○ If my sibling is male then he is my brother
  ○ hasBrother(?x,?y) ^ Male(?y) => hasBrother(?x,?y)
• OWL 2 extensions allow more rules expressible in OWL
  ○ Property chains (hasUncle example)
  ○ Datatype extensions
Explanations & Debugging

- Hard to understand large and/or complex ontologies
  - Why is this subclass relation inferred?
  - Why is this ontology inconsistent?
- Pellet provides axiom pinpointing service
  - For any inference, returns the (minimal set of) source axioms that cause the inference
  - Applications can track additional provenance info
- Axiom pinpointing is the first step to explanations
  - Lets users focus on significantly smaller parts
- Ongoing work on
  - Precise justifications (pinpoint parts of axioms)
  - Presentation of results

4/11/2008
Pellet Demo
Uses of Pellet

• Data integration problems
• Healthcare & Life Sciences (HCLS)
• Service oriented applications
• Policy analysis and verification
• Configuration management
• Probabilistic reasoning
Data Integration

• Querying over distributed and heterogeneous data sources
  ○ Describe data sources using ontologies
  ○ Define mappings between ontologies
  ○ Use reasoning to answer queries

• Some integration examples
  ○ Dynamic: NATO C3 Agency
  ○ Static: NASA
Healthcare & Life Sciences

• Rich and complex domains
  ◦ Terminology development and axiomatization
  ◦ Decision support
  ◦ Intelligent user interfaces
  ◦ Information integration

• Many large scale HCLS ontologies
  ◦ National Cancer Institute (NCI) Thesaurus
  ◦ Many other examples: SNOMED, GALEN, FMA, OBO, …
Service Oriented Applications

• Describe Web Services using ontologies
  ○ Input and output types
  ○ Pre and post-conditions
  ○ Languages: SA-WSDL, OWL-S, ...

• Reasoner supports tasks like
  ○ Matching requests with services
  ○ (Semi-)Automated composition of services
Policy Analysis (1)

• Policy languages: XACML, WS-Policy, etc.
  ◦ Languages for expressing policy constraints
• Managing policies is hard
  ◦ Detecting security holes
  ◦ Change impact analysis
• Tools focus on policy enforcement (run-time) not policy analysis (design-time)
Policy Analysis (2)

• Use OWL reasoning to analyze policies
  ◦ Map the policy language to OWL
  ◦ Done for XACML, and WS-Policy
• Analysis service provided
  ◦ Policy subsumption, redundancy, incompatibility, verification, querying
Configuration Management

• Find a set of components that satisfy a given system specification and constraints

• Many challenges
  ○ A decision about one component may have many unpredicted ramifications
  ○ Dependencies between components
Probabilistic Reasoning

• Many places for uncertainty
  ○ Taxonomic relationships can be uncertain
  ○ Facts can be uncertain
• Many application domains
  ○ Social network analysis
  ○ Breast cancer risk assessment
• More details in the Pronto presentation
Who are we?

• DC-based R&D firm specializing in Semantic Web, web services, and advanced AI technologies for federal and enterprise customers.

http://clarkparsia.com/