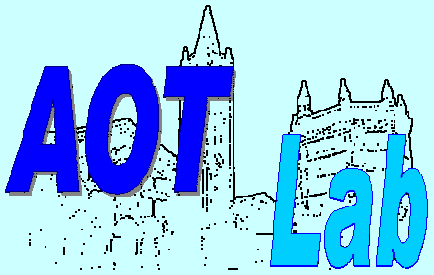
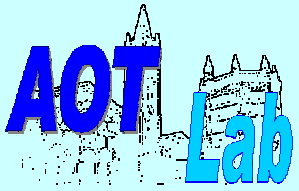


AOT Lab
Dipartimento di Ingegneria
dell'Informazione
Università degli Studi di Parma

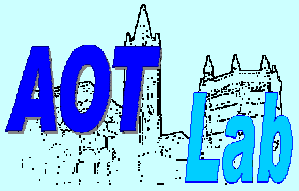
OWLBeans
From ontologies to Java classes

Michele Tomaiuolo
Federico Bergenti
Agostino Poggi
Paola Turci



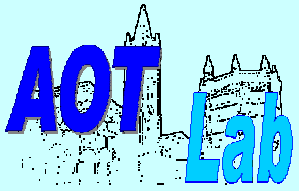
Summary

- Motivations
  - Semantic web
  - Multi-agent systems
  - Object-oriented systems
- Software architecture
  - Intermediate ontology model
  - Readers and writers
- Implementation
  - OWL ontologies
  - Java classes
  - JADE ontologies



Semantic aware agents

- Semantic web
  - Global network of machine-understandable information
  - OWL – Web Ontology Language
- Autonomous agents
  - Communication is a key feature
  - Messages convey information according to explicit or implicit ontologies
  - Need to access and manage information deployed on the semantic web

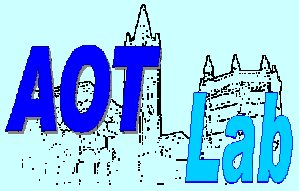


OWLBeans scope

- A link between these two worlds is necessary

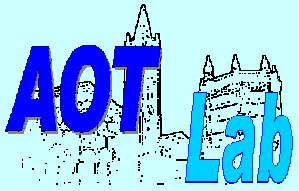
But not always agents need to (or can) manage the whole complexity of semantic web

- Taxonomies and simple relations
  - is-a
  - part-of



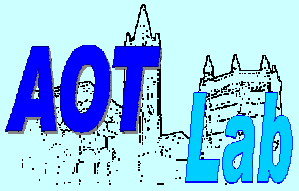
OWLBeans architecture

- Intermediate ontology model
  - Simplicity
  - Richness
  - Primitive data-types
  - External references
- Pluggable readers and writers
  - OWL ontologies
  - JavaBeans
  - JADE ontologies



Existing models

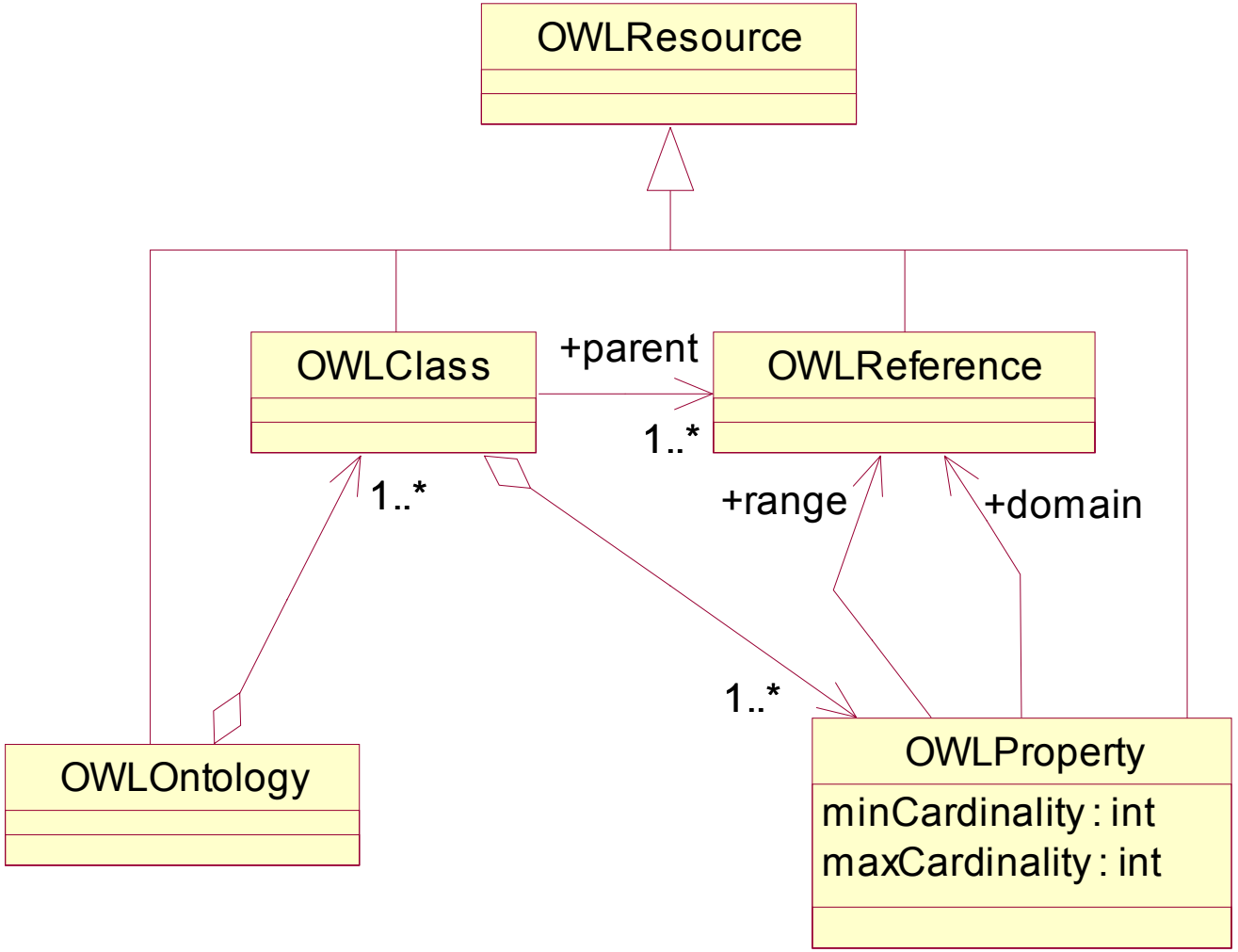
- Jena
  - OWL compliant
  - Too complex
  - Not handy for template engines

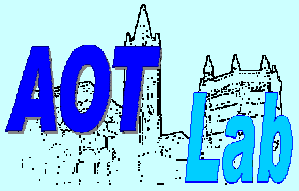


Existing models

- JADE
  - Simple
    - Handy form template engines
    - But no namespaces
  - FIPA oriented
    - Concepts
    - Predicates
    - Actions
  - Exploration
    - Classes
    - Parents
    - Cardinality

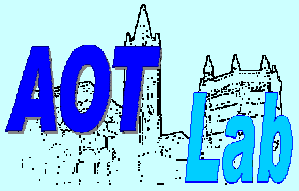
Intermediate ontology model





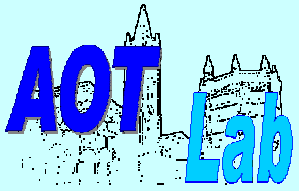
Reading OWL ontologies

- Limited subset of OWL
- Differences between the two worlds
  - Semantic web
  - Object-oriented systems
- Implementation based on Jena



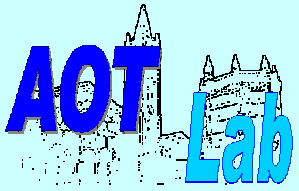
Writing OWL ontologies

- Easier problem but…
- Property names
  - Class scope in object-oriented systems
    - Unique only for their own class
  - Ontology scope in semantic web
    - Unique in the whole ontology
- Workarounds
  - *UnionClass* as domain
  - *minCardinality*, *maxCardinality* restrictions
  - *allValuesFrom* restrictions
- Still a problem if datatype/object



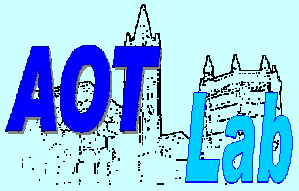
Mapping to/from OWL

OWL	OWLBeans
owl:Class	OwlClass
owl:ObjectProperty, owl:DatatypeProperty	OwlProperty
rdfs:range	OwlProperty.range
rdfs:domain	OwlProperty.domain
owl:FunctionalProperty	OwlProperty.maxCardinality
owl:minCardinality	OwlProperty.minCardinality
owl:maxCardinality	OwlProperty.maxCardinality
owl:cardinality	OwlProperty.minCardinality, OwlProperty.maxCardinality



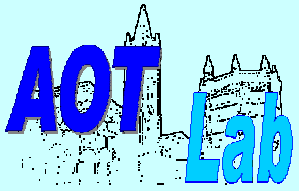
Template engine

- Available templates
  - JavaBeans
  - JADE ontologies
- Implementation based on Velocity
  - LGPL
  - Apache Group



Genarating JavaBeans

- Java packages
  - Classes of the same ontology
- Single inheritance issue
  - Interfaces and classes
- Cardinality is not checked
- Non-functional properties
  - `java.util.List`

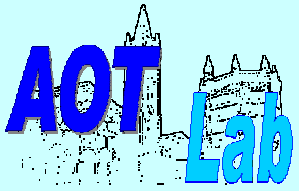


Mapping to/from Java

XSD	Java
xsd:boolean	boolean
xsd:decimal, xsd:float, xsd:double	double
xsd:integer, xsd:nonNegativeInteger, xsd:positiveInteger, xsd:nonPositiveInteger, xsd:negativeInteger, xsd:long, xsd:int, xsd:short, xsd:byte, xsd:unsignedLong, xsd:unsignedInt, xsd:unsignedShort, xsd:unsignedByte	int
xsd:base64Binary, xsd:hexBinary	Object
xsd:dateTime, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay, xsd:gMonth, xsd:duration	Date
xsd:string, xsd:normalizedString, xsd:anyURI, xsd:token, xsd:language, xsd:NMTOKEN, xsd:Name, xsd:NCName	String

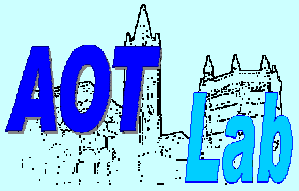
Generating JADE ontologies

- JADE agents can import OWL ontologies
  - JavaBeans for ontology classes
  - Subclass of `jade.content.onto.Ontology`
- ACL messages
  - Automatic marshalling and unmarshalling of JavaBeans



Mapping to/from JADE

OWLBeans	JADE
OwlClass	ObjectSchema
OwlProperty	SlotDescriptor
OwlProperty.range	SlotDescriptor.schema
OwlProperty.minCardinality	SlotDescriptor.optionality
OwlProperty.minCardinality	CardinalityFacet.cardMin
OwlProperty.maxCardinality	CardinalityFacet.cardMax
OwlProperty.range	TypedAggregateFacet.type



Mapping to/from JADE

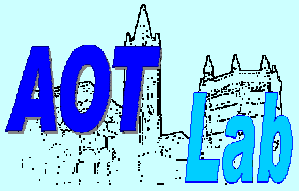
XSD	JADE
xsd:boolean	BOOLEAN
xsd:decimal, xsd:float, xsd:double	FLOAT
xsd:integer, xsd:nonNegativeInteger, xsd:positiveInteger, xsd:nonPositiveInteger, xsd:negativeInteger, xsd:long, xsd:int, xsd:short, xsd:byte, xsd:unsignedLong, xsd:unsignedInt, xsd:unsignedShort, xsd:unsignedByte	INTEGER
xsd:base64Binary, xsd:hexBinary	BYTE_SEQUENCE
xsd:dateTime, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay, xsd:gMonth, xsd:duration	DATE
xsd:string, xsd:normalizedString, xsd:anyURI, xsd:token, xsd:language, xsd:NMTOKEN, xsd:Name, xsd:NCName	STRING

Generating JADE ontologies

- Subclass of `jade.content.onto.Ontology`
- Can be instantiated at run-time

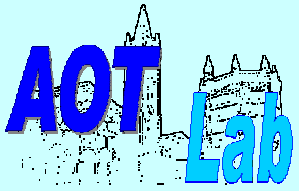
- JADE ontology model
  - Package `jade.content.onto`
  - Without generating any Java code
  - Without JavaBeans

- Java scripting engine + …
  - Java code for JADE ontology
  - Java code for JavaBeans (ontology classes)



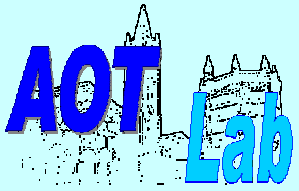
Scripting engine

- Exploited as a special class-loader
  - To load classes directly from Java source files
  - Without first compiling them into byte-code
- Janino / BeanShell
  - Janino manages multiple inheritance among interfaces
- An additional template is provided
  - Source of all interfaces, classes and JADE ontologies in a single stream for Janino
- JavaBeans can be loaded into the Java Virtual Machine directly from an OWL file



Why a scripting engine?

- Software agents for e-commerce
  - Trade goods and services described by a number of different, custom ontologies
- Agentcities network
  - Different basic services can be composed dynamically to create new compound services
- Ontology agnostic agents
  - To increase adaptability, these agents should be able to load ontology-specific classes and code at runtime
- OWLBeans
  - Load into the Java Virtual Machine some JavaBeans directly from an OWL file, together with the ontology-specific code needed to reason about the new concepts

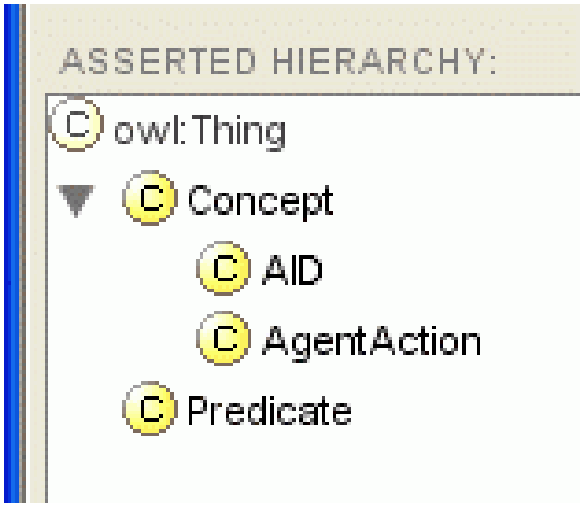


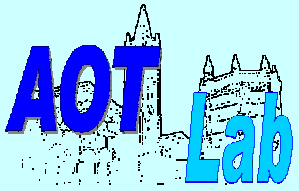
Reading JADE ontologies

- JADE ontologies not designed to be traversed from the outside
- They lack…
  - A method, in the `Ontology` class, to obtain the name of all defined classes
  - A method in the `ObjectSchema` class to get the name of all defined properties
  - Two methods to read minimum and maximum allowed cardinality, in `CardinalityFacet`
- Java reflection can be a temporary workaround

Reading JADE ontologies

- Basic FIPA classes are silently added to the ontology
  - Concept
  - AID
  - AgentAction
  - Predicate





Conclusions

- OWLBeans toolkit
  - Eases access to semantically annotated information by software agents
- Main functionality
  - Generating JavaBeans and other artefacts
  - Useful for agents needing just an object-oriented model of their application domain
- Modular design
  - The toolkit is able to process various kinds of input and produce different outputs

Questions?

- tomamic@ce.unipr.it