



## Ph.D. course: Advanced Distributed Systems Development with Multiagent Systems

Dott. Daniela Briola

Prof.ssa Viviana Mascardi

Prof. Rafael Bordini





## JADE: Ontologies

Daniela Briola
Daniela.briola@unimib.it

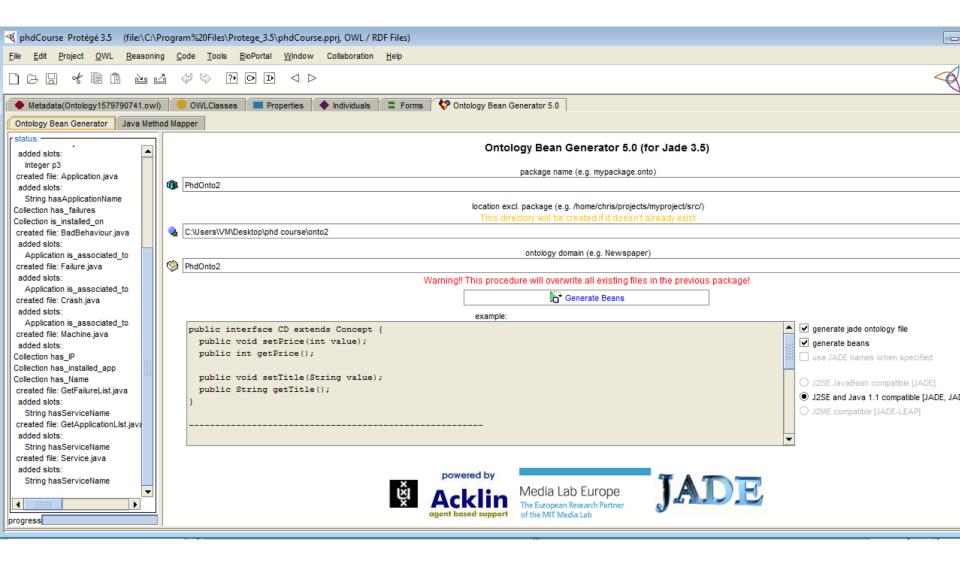
Ph.D. Course: Advanced Distributed Systems Development with Multiagent Systems

## OBJ5 (based on OBJ4)

- It creates a Java version of the OWL ontology (with some limitations due to the different expressivity of them)
- A predefined ontology model the Jade foreseen Concepts
  - All the new classes must be subclasses of the Jade Concepts
- For each class, an interface is created
- Subclass relationship is mapped into "extends" between interfaces
- Multiple properties are mapped into List
- Functional properties are mapped into simple fields
  - Cardinality is not translated
  - Object type is translated using the Java interfaces as types
- A Default implementation of each OWL Class is generated, implementing the corresponding interface
- A BeanOntology class is created to manage in JADE all the concepts and properties
- A jar to be used in Jade is created

## Let's try

- Download the new version of the ontology we developed together
- Have a look to the new classes (C1... Services) and their properties (in particular, the "inverse of")
- Import (again) the Ontology of Jade (do not change its name, only reimport it) and save the project
- Under Project → Configure, add the Tab of OntologyBeanGenerator
- Export the ontology as in figure (next slide)
- Download the Mas2 from the course site
- Update the BuildPath to include the ontology



- Have a look to the code:
  - See how the ontology is used to send and receive messages based on the ontology
  - See how agents register services and search for them
- Create 3 agents of type AgentService (call them as you wish)
- Start the sniffer, sniffing them
- Start a AgentRequester, and check the results on the sniffer
- Try to uncomment line 67 in AgentService...