



G I S U M



UNIVERSIDAD
DE MÁLAGA

GISUM

Towards the automatic derivation of Malaca agents using MDE

Inmaculada Ayala Viñas, Mercedes Amor Pinilla and
Lidia Fuentes Fernández
{ayala, pinilla, lff}@lcc.uma.es

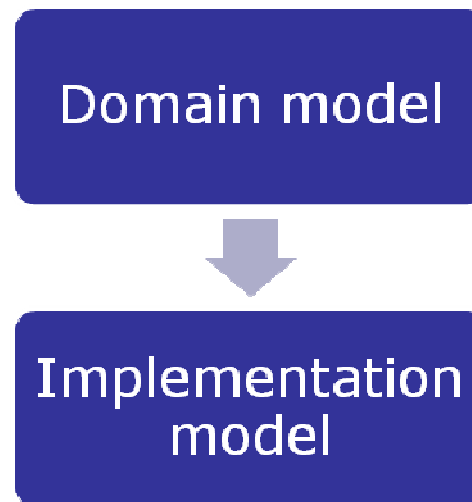
Grupo CAOSD

<http://caosd.lcc.uma.es/>

AOSE 2010
Toronto, May

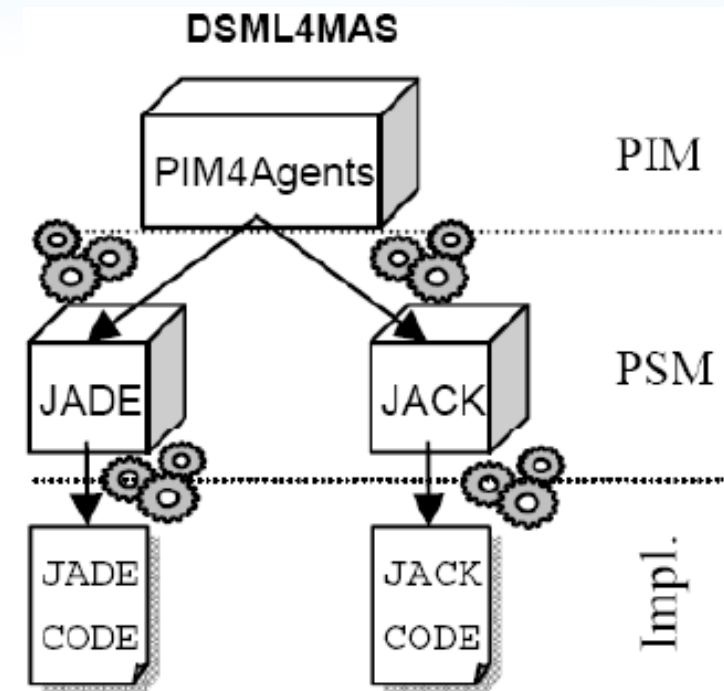
- ☐ Introduction
- ☐ From PIM4Agents to Malaca
- ☐ Discussion
- ☐ Conclusion

- Our motivation for this work is to bridge the gap between design and implementation of agents.
 - Domain concepts at high level of abstraction.
 - Generation more or less automatically implementations from design.
- Model Driven Engineering (MDE)
 - To specify a Multi-Agents System (MAS) in a platform independent metamodel (PIM).
 - To transform automatically to a platform specific metamodel (PSM).



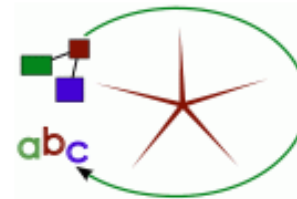
- ▣ Several recent works uses MDE ideas to enhance agent development process.

- ▣ Domain Specific Modeling Language for MAS (DSML4MAS)
 - > PIM4Agents
 - > M2M and M2T transformations
 - JACK
 - JADE
 - > DSML4MAS Development Environment (DDE) Tool



- ❏ What is the cost of including a new AP in this proposal?
 - ❏ Define a new M2M transformation rules (to the target metamodel of the new AP)
 - ❏ Define a new M2T transformation rules (to code provided by the API of the new AP)

- ❏ This is a very complex task:
 - ❏ The metamodel of the target platform must be available.
 - ❏ Sometimes the target metamodel is not specified completely.
 - ❏ An expert in transformation languages
 - ❏ Deep knowledge of the target implementation framework.



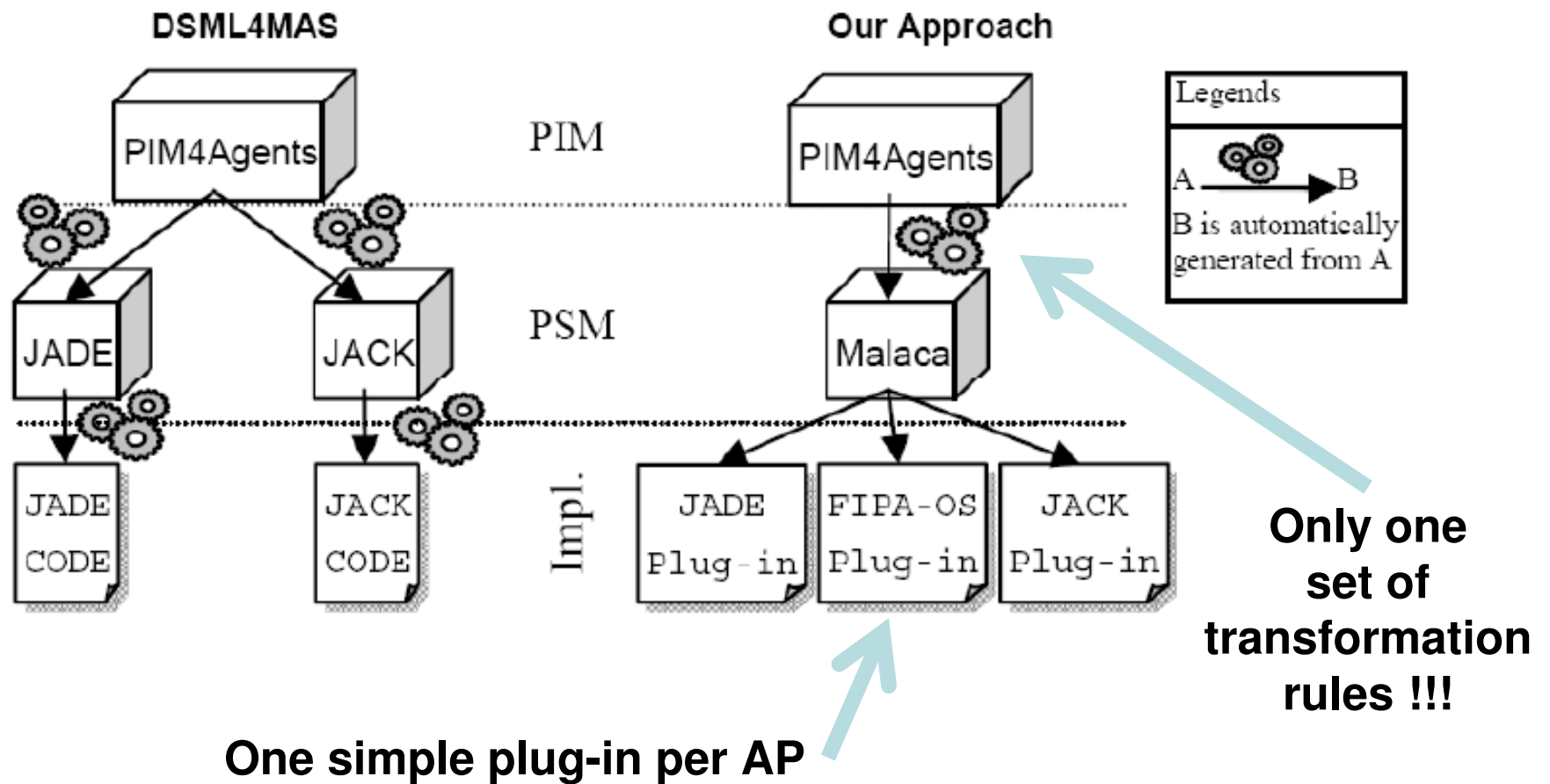
- ☐ To simplify the transformation process, we propose to transform PIM4Agent to Malaca.

- ☐ PIM4Agents as PIM
 - ☐ It is possible to represent concepts from different agent types.
 - ☐ It is easy to specify MAS for different domains.
 - ☐ The DDE Tool helps to specify different views of MAS.

- ☐ Malaca as PSM
 - ☐ A platform-neutral specific metamodel (PNSM)
 - ☐ Execution on top of any FIPA compliant platforms
 - ☐ MAD Tool

- ☐ Introduction
- ☐ **From PIM4Agents to Malaca**
- ☐ Discussion
- ☐ Conclusion

From PIM4Agents to Malaca



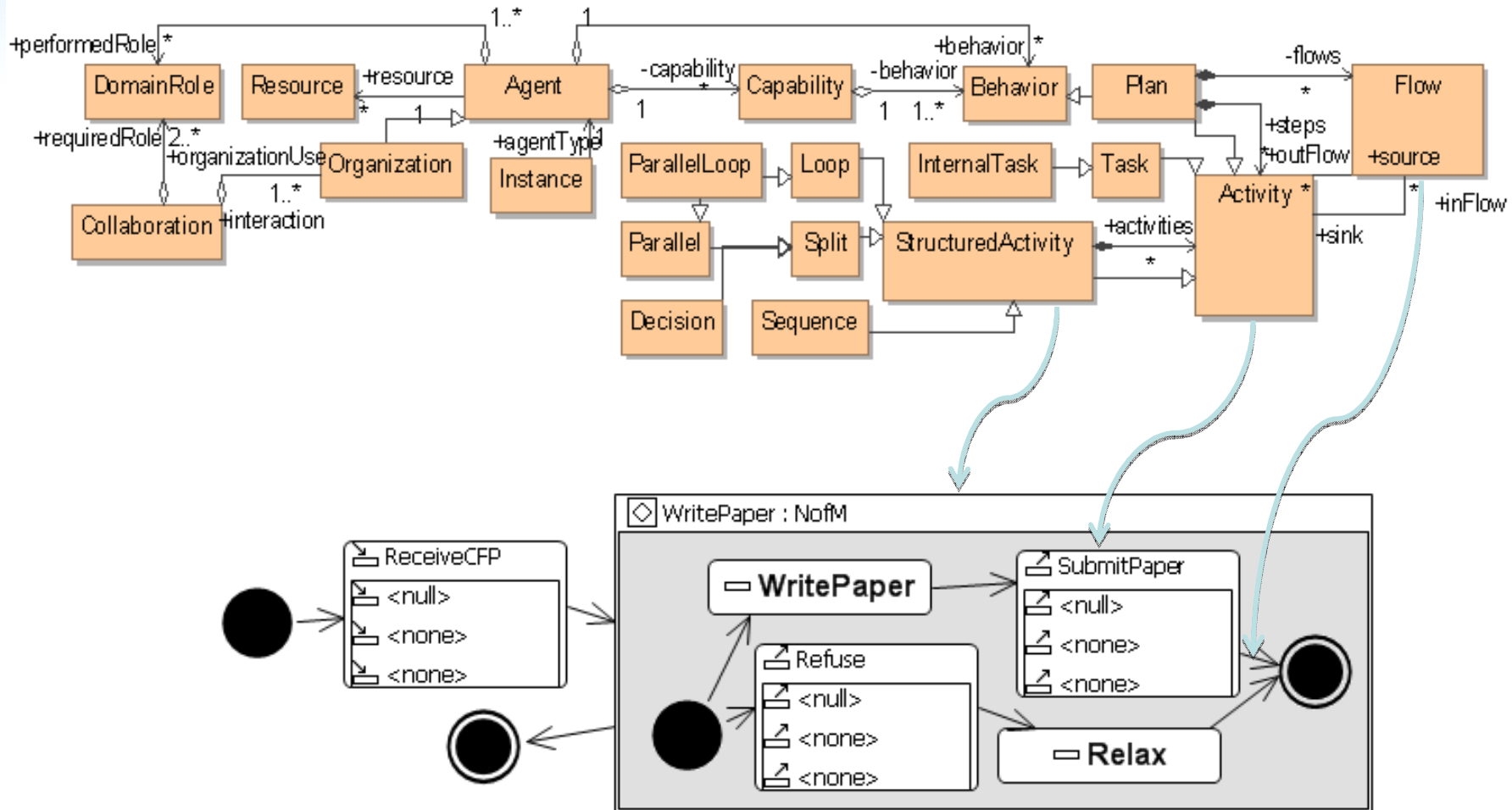
- The internal architecture of a Malaca agent represents separately application-specific functions from extra-functional agent properties
 - Better internal modularization
 - Based on the composition of components and *aspects* (e.g. distribution aspect)

- Malaca is based in two XML-based domain specific languages.
 - MaDL
 - Description of the agent architecture in view of the agent functionality and the agent interaction
 - ProtDL
 - Interaction protocol

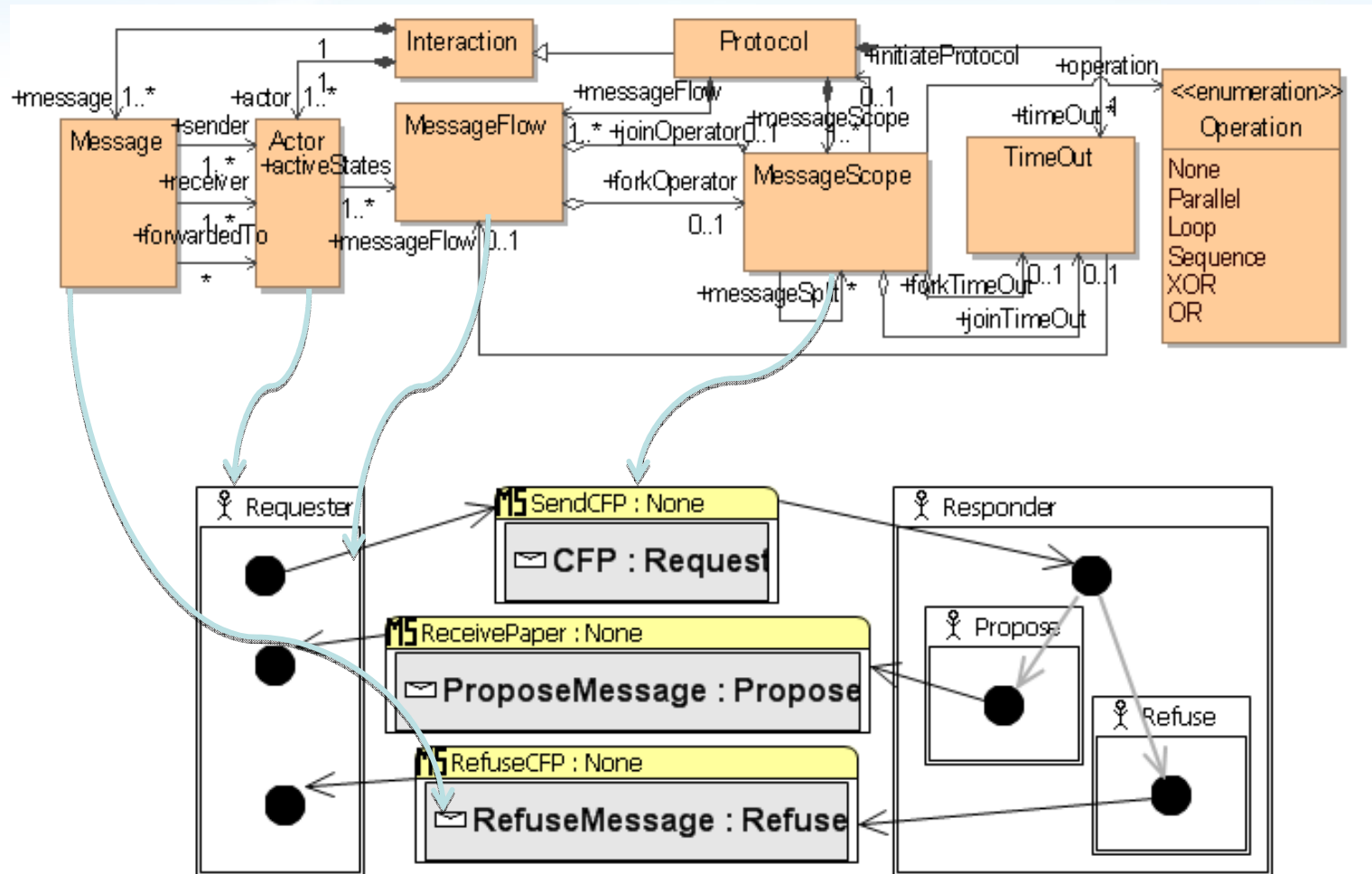
Transformation rules

	PIM4Agents	Malaca
MR1	Protocol	Protocol
MR2	Actor	RoleDescription
MR3	MessageFlow, MessageFlow	StateTransitionRule
MR4	MessageFlow, MessageFlow	TransitionDescription
MR5	Plan, String	RoleDescription
MR6	Activity, Activity	StateTransitionRule
MR7	Activity, Activity	TransitionDescription
MR8	InternalTask	ProcessComponent
MR9	Split	ProcessComponent
MR10	Protocol, Organization	Protocol

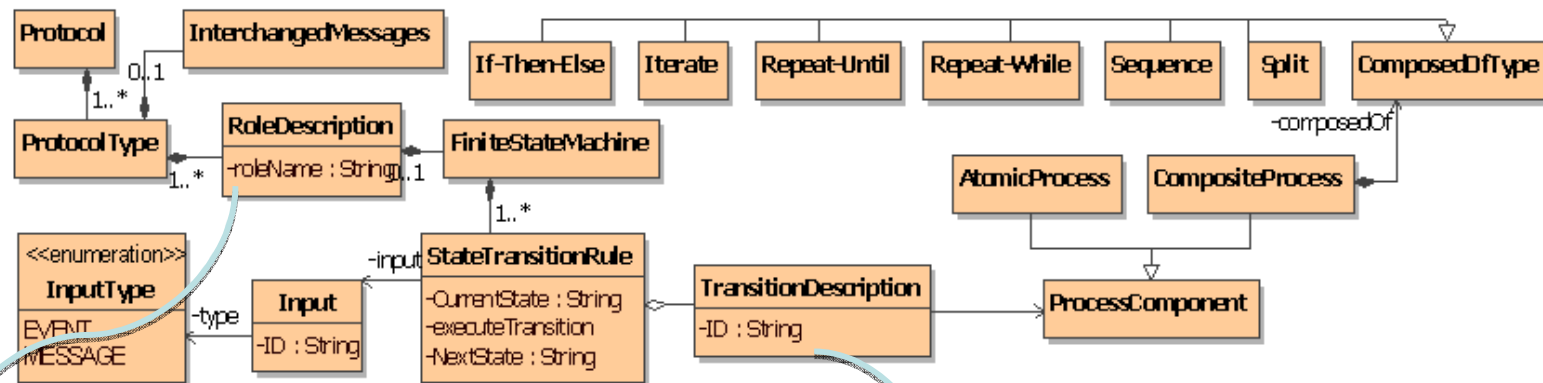
Agent viewpoint



Interaction viewpoint



ProtDL



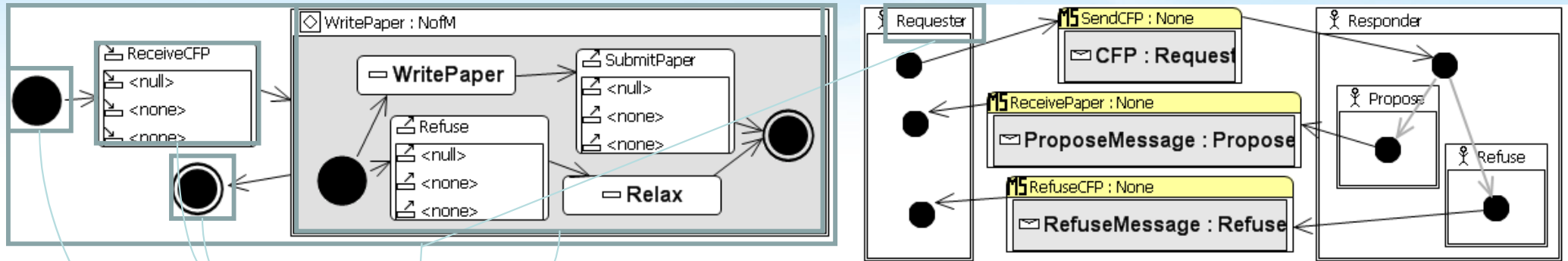
```

<roleDescription roleName="Responder">
  <finiteStateMachine>
    <stateDeclaration>
      <state>Begin</state>
      <state>End</state>
      <state>ReceiveCFP</state>
      <state>WritePaper</state>
      <finalState>End</finalState>
    </stateDeclaration>
    <stateTransitionRule currentState="Begin" nextState="WritePaper" name="STRule Begin to WritePaper">
      <input value="value" id="InputType STRule Begin to WritePaper" type="MESSAGE"/>
      <executeTransition value="value" id="IdentifierType STRule Begin to WritePaper"/>
    </stateTransitionRule>
    <stateTransitionRule currentState="WritePaper" nextState="End" name="STRule WritePaper to End">
      <input value="value" id="InputType STRule WritePaper to End" type="EVENT"/>
      <executeTransition value="value" id="IdentifierType STRule WritePaper to End"/>
    </stateTransitionRule>
  </finiteStateMachine>
</roleDescription>
  
```

```

<transitionDescription id="IdentifierType STRule Begin to SendCFP">
  <process>
    <atomicProcess>
      <sendMessage messageId="SendCFP"/>
    </atomicProcess>
  </process>
</transitionDescription>
  
```

From PIM4Agents to Malaca



MR6,
MR7

MR5

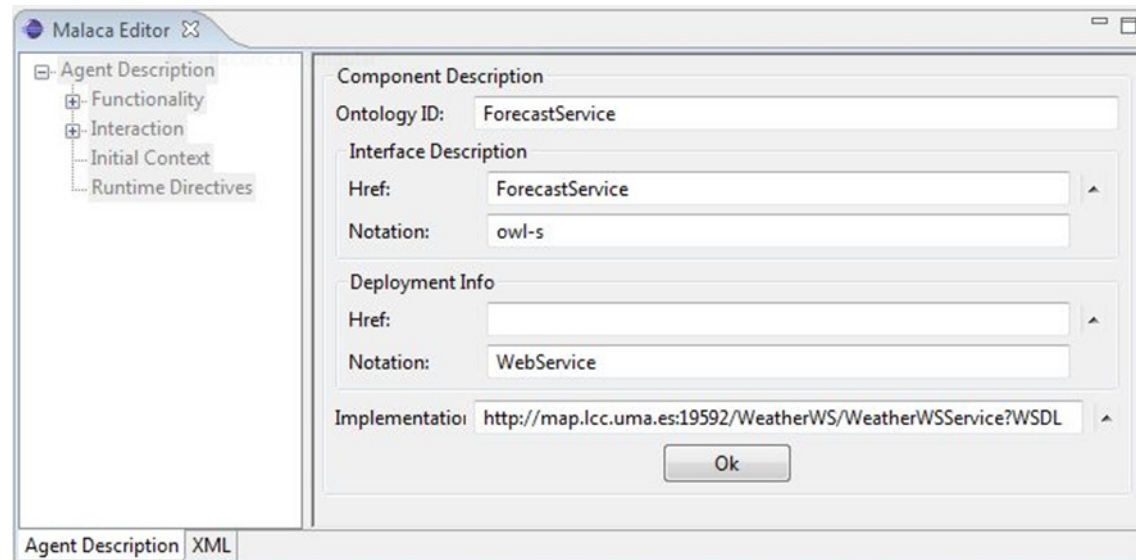
MR10

```

<roleDescription roleName="Responder">
  <finiteStateMachine>
    <stateDeclaration>
      <state>Begin</state>
      <state>End</state>
      <state>ReceiveCFP</state>
      <state>WritePaper</state>
      <finalState>End</finalState>
    </stateDeclaration>
    <stateTransitionRule currentState="Begin" nextState="WritePaper" name="STRule Begin to WritePaper">
      <input value="value" id="InputType STRule Begin to WritePaper" type="MESSAGE"/>
      <executeTransition value="value" id="IdentifierType STRule Begin to WritePaper"/>
    </stateTransitionRule>
    <stateTransitionRule currentState="WritePaper" nextState="End" name="STRule WritePaper to End">
      <input value="value" id="InputType STRule WritePaper to End" type="EVENT"/>
      <executeTransition value="value" id="IdentifierType STRule WritePaper to End"/>
    </stateTransitionRule>
  </finiteStateMachine>
</roleDescription>
  
```

- ☐ Introduction
- ☐ From PIM4Agents to Malaca
- ☐ **Discussion**
- ☐ Conclusion

- ❏ Some PIM4Agents concepts cannot be mapped to Malaca.
 - Organization
- ❏ Some Malaca concepts cannot be mapped to PIM4Agents
 - Functionality



Generated JADE code

- For 1 protocol and 2 plans the DDE Tool generates 22 java classes.
- MAS structure
 - Functionality added as standard Java code.
- Code is not optimized
 - FIPA protocol templates

- ☐ Introduction
- ☐ From PIM4Agents to Malaca
- ☐ Discussion
- ☐ **Conclusion**

- ❑ This paper presents a MDE approach to develop MAS using PIM4Agents as PIM and Malaca as PSM.
- ❑ We have defined mapping rules to generate a set of MaDL/ProtDL files that can be used to deploy and execute Malaca agents.
- ❑ The derivation of agents for new APs is enhanced in this approach, since it is accomplished by simply selecting the appropriate distribution aspects.

Thank you for your attention