Escola Tècnica Superior d'Enginyeria
UNIVERSITAT ROVIRA I VIRGILI

[DΣIM]

Multi-Agent Systems
Jordi Pascual – jordi.pascual@urv.cat

Interaction protocols

MESIIA – Master's Degree in Computer Security Engineering and Artificial Intelligence
MAI - Master's Degree in Artificial Intelligence

Outline

1. Protocols

2. AchieveRE Initiator / Responder

3. Auctions

4. FIPA-Contract Net

5. Protocols in Dedale

6. More resources

1. Protocols

- Standard interaction protocols specified by FIPA
- Standard templates to build agent conversations
- Eleven different protocols available in JADE
- Most used:
  - **AchieveRE Initiator / Responder**
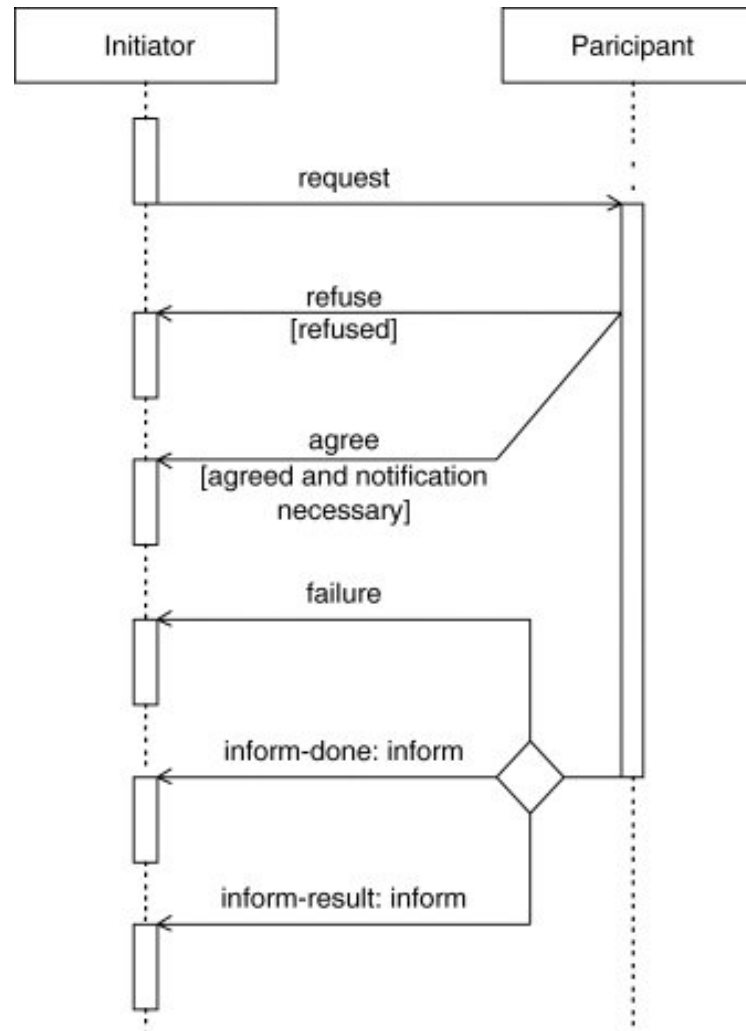 - **FIPA-Contract Net**

2. AchieveRE Initiator / Responder

- Achieve **Rational Effect** (i.e. The reason why the message is sent)

- **Initiator**: sends a request to the Responder

- **Responder**: replies to the request with *not-understood/refuse/agree*. If it does agree, must send the results of the RE (*failure/inform done/inform results*)
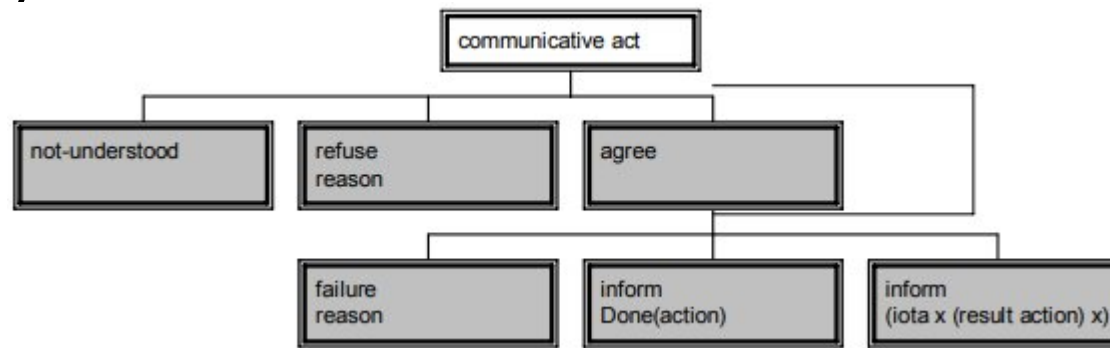
2. AchieveRE Initiator / Responder

Message flow

2. AchieveRE Initiator / Responder

Two types of RE Initiator / Responder are already implemented in JADE

SimpleAchieveRE Initiator/Responder

- App-specific Behaviours cannot be added

- Communication **1:1**

- An Initiator handles a single Responder
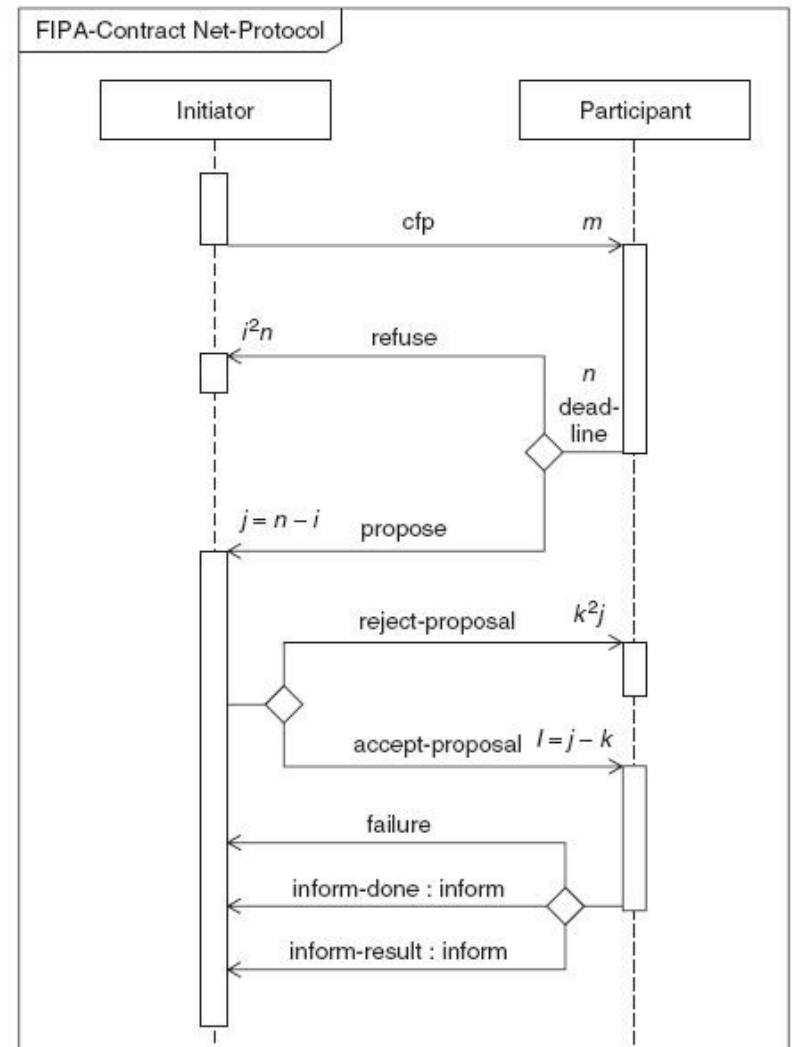
AchieveRE Initiator / Responder

- Can be extended to add specific functionality

- Communication **1:N**

- An Initiator can handle multiple Responders

3. Auctions

- **English** and **Dutch** auctions not implemented in JADE
- Some ideas to manually implement them ordered by **quality**
  - **FSM Behaviour**: protocol states can be mapped to the states on the FSM
  - **Several Behaviours**: each agent is responsible for a part of the protocol. Consider using Message Templates
  - **Single Behaviour**: all the cases and states are managed by a single behaviour

4. FIPA-Contract Net

- Interaction protocol based on proposals

- More specialized than AchieveRE

- Steps:
  1. Initiator sends a **Call For Proposal** (CFP) to a **set of responders**, including a set of preconditions (optional)
  2. The responders receive the **CFP**, evaluate the proposals and may:
     1. **Accept** the proposal
     2. **Reject** the proposal
  3. The initiator evaluates the proposals and **accepts** one of them and **rejects** the others

FIPA-Contract Net-Protocol

| Initiator | | Participant |

cfp m

i^2n refuse

n dead-line

$j = n - i$ propose

reject-proposal k^2j

accept-proposal $l = j - k$

failure

inform-done : inform

inform-result : inform

4. FIPA-Contract Net

In order to use the already implemented Contract Net, some methods have to be overridden to add the desired functionality

ContractNetInitiator

- handlePropose(): called when one **proposal** message is received

- handleRefuse(): called when one **refuse** message is received

- handleAllResponses(): called when **ALL** the responses are collected. From here we can determine the best proposal

- handleInform(): called once an **inform** message is received

- handleFailure(): called when one **failure** message is received

ContractNetResponder

- handleCfp(): handles the **CFP** and replies with a proposal or a refusal. According to the decision (**accept** or **reject**), one of these two methods is invoked afterwards

- handleAcceptProposal(): sends the **accept** message for the given CFP

- handleRejectProposal(): sends the **reject** message for the given CFP

4. FIPA-Contract Net: Example

- We will run the example Contract Net protocol from the JadeExample project

- We are interested in the **ContractNetInitiatorAgent** and the **ContractNetResponderAgent** of the *urv.imas.protocols* package

- We will manually create the agents for this simple example

- Start the JADE GUI using the **jade-gui** profile (*mvn install -P jade-gui exec:java*) and follow the next steps

4. FIPA-Contract Net: Example

1. Manually add a **ContractNetResponderAgent** named **responder** into the **Main-Container**

2. The responder will wait for a **Call For Proposal** (CFP) from the **ContractNetInitiatorAgent**
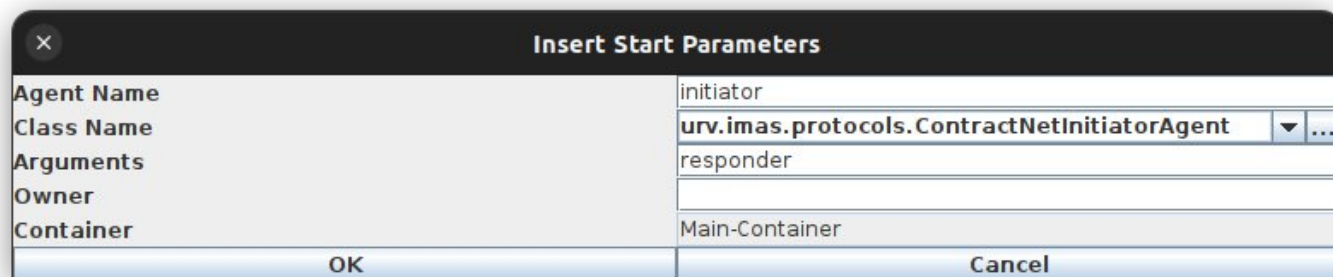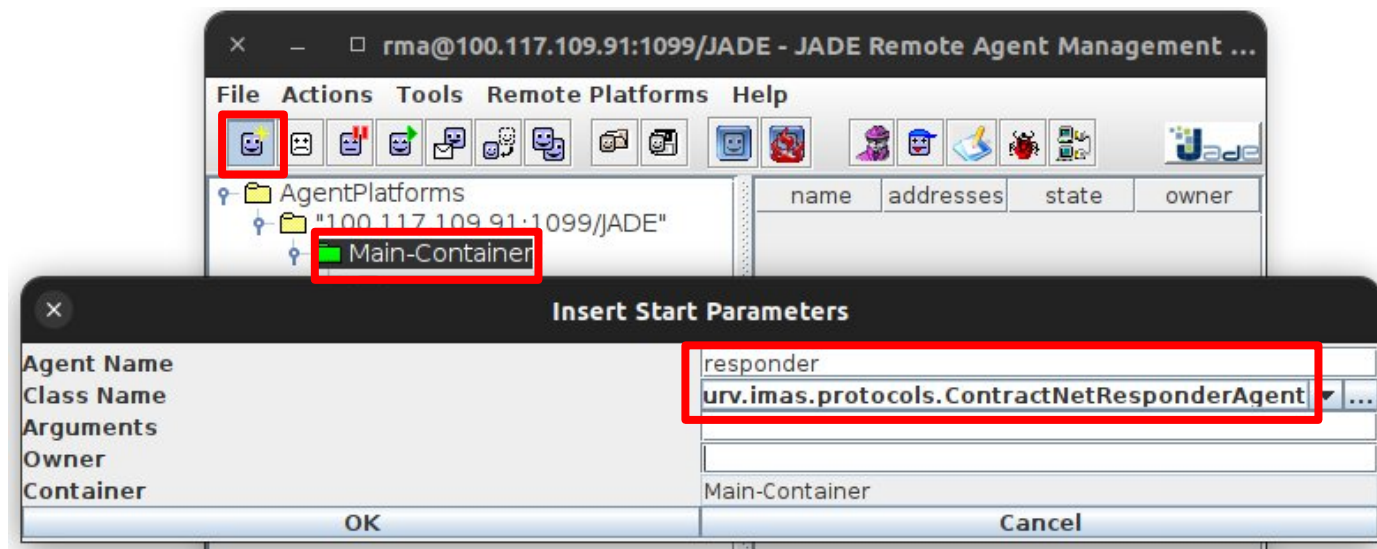
3. Check the **ContractNetInitiatorAgent** code. It needs as an argument the AIDs (names) of the responder agents

```
// Read names of responders as arguments
Object[] args = getArguments();
if (args != null && args.length > 0) {
        nResponders = args.length;
        System.out.println("Trying to delegate dummy-action to one out of " + nResponders + "
responders.");
        ...
}
```

4. Add a new **ContractNetInitiatorAgent** named **initiator** to the **Main-Container**. Set **responder** as the argument

Insert Start Parameters	
Agent Name	initiator
Class Name	urv.imas.protocols.ContractNetInitiatorAgent ▼ …
Arguments	responder
Owner	
Container	Main-Container
OK	Cancel

4. FIPA-Contract Net: Example

5. When the Initiator loads, the **Call For Proposal** is sent to the Responder. In the command prompt or shell you can observe the interaction between the agents

```
Trying to delegate dummy-action to one out of 1 responders.
Agent responder: CFP received from initiator@10.5.0.2:1099/JADE. Action is dummy-action
Agent responder: Proposing 5
Agent responder@10.5.0.2:1099/JADE proposed 5
Accepting proposal 5 from responder responder@10.5.0.2:1099/JADE
Agent responder: Proposal accepted
Agent responder: Action successfully performed
Agent responder@10.5.0.2:1099/JADE successfully performed the requested action
```

4. FIPA-Contract Net: Exercise

Follow the next steps:

1. Check the **ContractNetInitiatorAgent** and **ContractNetResponderAgent** code, and guess what they do

2. Create a new Maven profile which opens the JADE GUI and starts at least **3 Responder Agents**

3. Build and run the new Maven profile

4. Using the JADE GUI, create a new **Initiator** agent and start the Contract Net protocol with all the Responder agents

Optional: use the sniffer to see the communication between agents

4. FIPA-Contract Net: Exercise solution

1. The Responder randomly decides if it accepts or refuses the CFP (see **handleCfp()**) and sends the random number as the proposal. The initiator selects the proposal with the higher number (see **handleAllResponses()**). The Responder randomly decides if the proposal is performed, or it fails (see **handleAcceptProposal()**)

2. The profile should be something like
   1. ID: **jade-responders**
   2. Argument: **-gui**
   3. Argument: **-agents**
   4. Argument: **responder1:urv.imas.protocols.ContractNetResponderAgent; responder2:urv.imas.protocols.ContractNetResponderAgent; responder3:urv.imas.protocols.ContractNetResponderAgent**

4. FIPA-Contract Net: Exercise solution

3. Build and run the new profile using **mvn install -P jade-responders exec:java**

```
Agent responder2 waiting for CFP...
Agent responder1 waiting for CFP...
Agent responder3 waiting for CFP...
```

Optional: start the sniffer, and do sniff the 3 responder agents

4. Manually create the Initiator agent

Insert Start Parameters	
Agent Name	initiator
Class Name	urv.imas.protocols.ContractNetInitiatorAgent
Arguments	responder1,responder2,responder3
Owner	
Container	Main-Container
OK	Cancel

4. FIPA-Contract Net: Exercise solution

5. Check the obtained results by the Contract Net

```
Trying to delegate dummy-action to one out of 3 responders.
Agent responder1: CFP received from initiator@10.5.0.2:1099/JADE. Action is dummy-action
Agent responder2: CFP received from initiator@10.5.0.2:1099/JADE. Action is dummy-action
Agent responder1: Proposing 9
Agent responder2: Refuse
Agent responder3: CFP received from initiator@10.5.0.2:1099/JADE. Action is dummy-action
Agent responder3: Proposing 9
Agent responder1@10.5.0.2:1099/JADE proposed 9
Agent responder2@10.5.0.2:1099/JADE refused
Agent responder3@10.5.0.2:1099/JADE proposed 9
Accepting proposal 9 from responder responder1@10.5.0.2:1099/JADE
Agent responder1: Proposal accepted
Agent responder1: Action execution failed
Agent responder3: Proposal rejected
Agent responder1@10.5.0.2:1099/JADE failed
```

5. Protocols in Dedale

1. When using message exchange, you must use the *sendMessage* method from Dedale instead of JADE's *send*

2. The **DedaleAchieveREInitiator** and **DedaleContractNetInitiator** classes available on the URV Virtual Campus (*Dedale Protocols.zip*) must be used instead of the JADE's ones

3. The JADE implementation of **AchieveREResponder** and **ContractNetResponder** can be used

6. More resources

- [FIPA Protocols](#)

- [JADE Javadoc](#)

- [JADE Guides](#)

- [JADE Maven Setup for Beginners](#)