Multi-Agent interactions

Wassila Ouerdane & Nicolas Sabouret

CentraleSupélec

3 novembre 2018

Contents

Interaction mechanisms

Indirect interactions

Direct interactions
- Speech Acts Theory
- Message structure

How does this work ?

Interaction protocols

Conclusion

Asynchronous interactions

Problem

- Agents run asynchronously
- Method invocation is synchronous

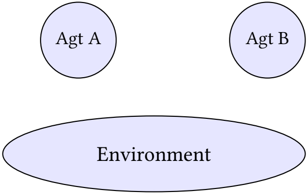
Asynchronous interactions

Problem

- Agents run asynchronously
- Method invocation is synchronous

PRS Architecture

- Actions modify the environment
- (Asynchronous) perception of the modification



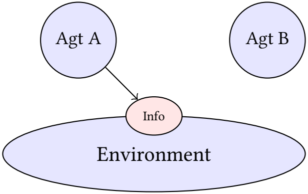
Asynchronous interactions

Problem

- Agents run asynchronously
- Method invocation is synchronous

PRS Architecture

- Actions modify the environment
- (Asynchronous) perception of the modification



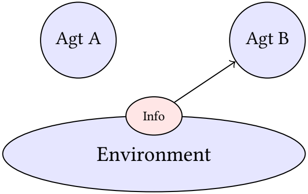
Asynchronous interactions

Problem

- Agents run asynchronously
- Method invocation is synchronous

PRS Architecture

- Actions modify the environment
- (Asynchronous) perception of the modification



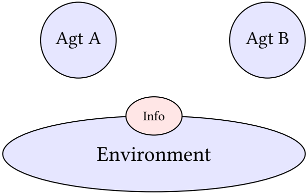
Asynchronous interactions

Problem

- Agents run asynchronously
- Method invocation is synchronous

PRS Architecture

- Actions modify the environment
- (Asynchronous) perception of the modification



Indirect interactions

Principle

- No intention to communicate to a specific agent
- Modification of the environment perceived by others

Indirect interactions

Principle

- No intention to communicate to a specific agent
- Modification of the environment perceived by others

Architectures

- Blackboard : shared centralized ressource, free modification (1990)

Indirect interactions

Principle

- No intention to communicate to a specific agent
- Modification of the environment perceived by others

Architectures

- Blackboard : shared centralized ressource, free modification (1990)
- Geographical situatedness (1995)
  - Environment = grid of cells that can be modified
  - Agents move to the positions they need to observe
  - Also used for simulation
- Stigmergy (1997)
  - Cell content modified without agent action
  - Global modification that can use nearby cells
  - Simulation of ant colonies

Agents and artifacts (2003)

Alexandro Ricci

Principle

Indirect interactions through specific objects (artifacts)

Agents and artifacts (2003)

Alexandro Ricci

Principle

Indirect interactions through specific objects (artifacts)

Artifact

- Stores data (like agents do…)
- Has a set of invocation methods (like web services…)
- No autonomy ! → No possibilty to refuse a request
- KR-based representation of operations for agents to reason upon
- Process data asynchronously (like agents do…)

Agents and artifacts (2003)

Alexandro Ricci

Principle

Indirect interactions through specific objects (artifacts)

Artifact

- Stores data (like agents do. . .)
- Has a set of invocation methods (like web services. . .)
- No autonomy ! → No possibilty to refuse a request
- KR-based representation of operations for agents to reason upon
- Process data asynchronously (like agents do. . .)

Agents

- Ask for data processing by an artifact
- Register to receive an alert when the operation is done
- Can register to monitor a communication artefact

Direct interactions

Principle

- Intention to communicate to a specific agent
  - Messages with sender, recipient and structured content

Direct interactions

Principle

- Intention to communicate to a specific agent
  - Messages with sender, recipient and structured content

Problem : heterogeneity

Communication layers :

- Transport level → environment
- Syntax level → message structure
- Semantic level → knowledge representation
- Pragmatic level → protocols

⇒ This has to be normalised !

Contents

Interaction mechanisms

Indirect interactions

Direct interactions

- Speech Acts Theory
- Message structure

How does this work ?

Interaction protocols

Conclusion

Speech Acts Theory I

Searle, 1969

Communication is an action

Communicate → change interlocutor's mental state

Speech Acts Theory I

Searle, 1969

Communication is an action

Communicate → change interlocutor's mental state

Three aspect of a speech act

- Locutory : the act of saying
- Illocutory : the intention of the speech
- Perlocutory : the effect of the speech

Speech Acts Theory I

Searle, 1969

Communication is an action

Communicate → change interlocutor's mental state

Three aspect of a speech act

- Locutory : the act of saying
- Illocutory : the intention of the speech
- Perlocutory : the effect of the speech

Illocutory act

Performative(Content)

Examples : *Assert(rain), Order(rain), Question(rain)*, . . .

Speech Acts Theory II

Different researchers, different theories…

Searle

- Assertive acts : *Facts*
- Directive acts : *Actions* + *Questions*
  - « Do the action to tell about the Face »
- Promissive acts : *Commitments*
- Expressive acts : *Emotions*
- Declarative acts : *Protocols*

Speech Acts Theory II

Different researchers, different theories...

Searle

- Assertive acts : *Facts*
- Directive acts : *Actions + Questions*
  - « Do the action to tell about the Face »
- Promissive acts : *Commitments*
- Expressive acts : *Emotions*
- Declarative acts : *Protocols*

Sperber & Wilson

- Say that : assertions and promisses
- Say about : orders
- Ask if : questions

Speech Acts Theory II

Different researchers, different theories…

Searle

- Assertive acts : *Facts*
- Directive acts : *Actions + Questions*
  - « Do the action to tell about the Face »
- Promissive acts : *Commitments*
- Expressive acts : *Emotions*
- Declarative acts : *Protocols*

Sperber & Wilson

- Say that : assertions and promisses
- Say about : orders
- Ask if : questions

⇒ Need to define the **semantics** of each performative !

Contents

Interaction mechanisms

Indirect interactions

Direct interactions

- Speech Acts Theory
- Message structure

How does this work ?

Interaction protocols

Conclusion

FIPA-ACL standard

`http://www.fipa.org`

Message structure

- Recipient(s) = list of agent IDs
  - *The environment must provide each agent with a unique ID*
- Performative = 1 value in a list of predefined possible acts
- Content expressed in any knowledge representation language
  - *First Order Logics, Lisp syntax, …*

FIPA-ACL standard

`http://www.fipa.org`

Message structure

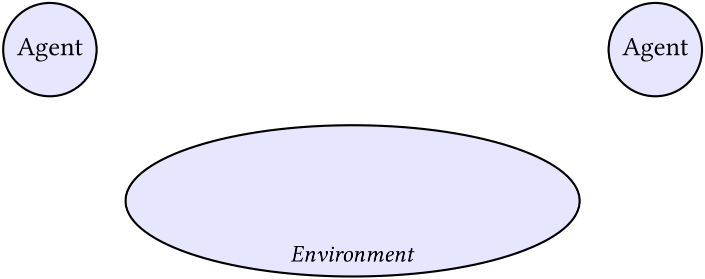
- Recipient(s) = list of agent IDs
  - *The environment must provide each agent with a unique ID*
- Performative = 1 value in a list of predefined possible acts
- Content expressed in any knowledge representation language
  - *First Order Logics, Lisp syntax, …*

Usual representation : $\langle snd, perf(rcv, content)\rangle$

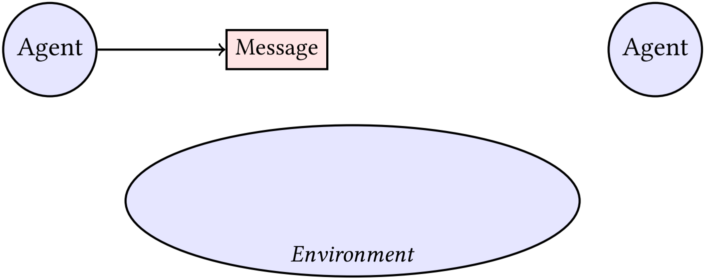
Communication management elements

- Sender
- Name of the content language
- Ontology referred to for Knowledge Representation
- Conversation ID and number of the message in the conversation

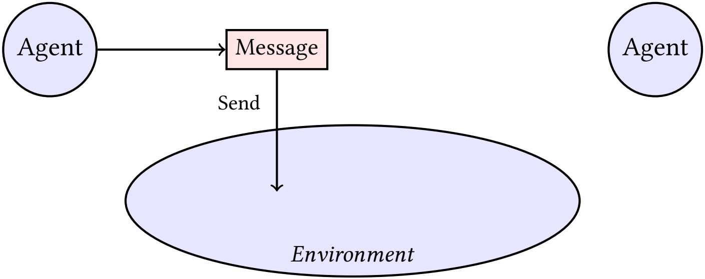
Asynchronous message sending



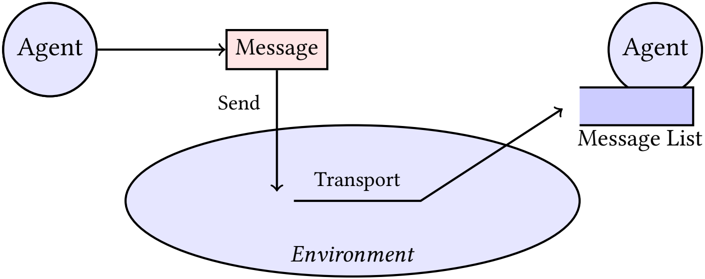
Asynchronous message sending



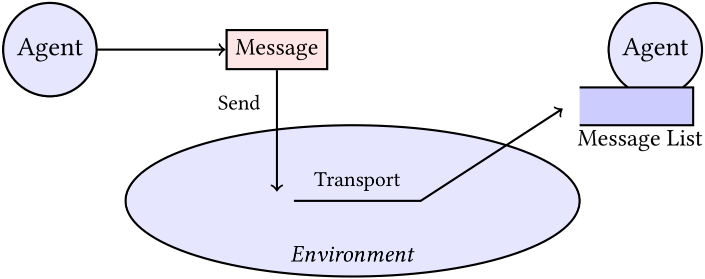
Asynchronous message sending



Asynchronous message sending

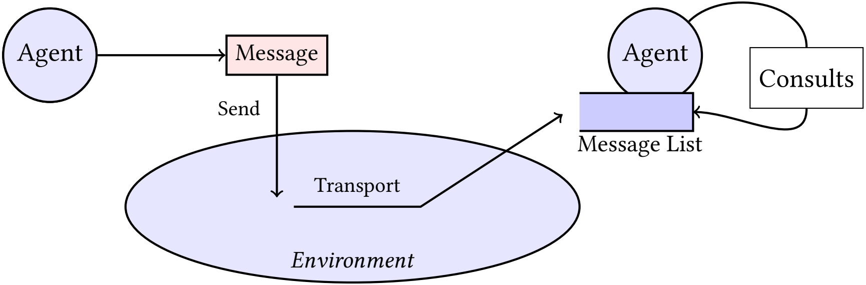


Asynchronous message sending



Synchronous method calls

Asynchronous message sending



Synchronous method calls	Asynchronous

Environement's role

Agent Management Service (AMS)

- Loading agents in the MAS
- Allocation of Agent IDs
- Management of the agents' list
- Running threads
- Message sending

Environement's role

Agent Management Service (AMS)

- Loading agents in the MAS
- Allocation of Agent IDs
- Management of the agents' list
- Running threads
- Message sending

In case of multi-platforms :

- Brokers for message sending
- Mobile agents : threads cloning, starting, ending…

Environement's role

Agent Management Service (AMS)

- Loading agents in the MAS
- Allocation of Agent IDs
- Management of the agents' list
- Running threads
- Message sending

In case of multi-platforms :

- Brokers for message sending
- Mobile agents : threads cloning, starting, ending...

Directory Facilitator (DF)

Yellow pages...

- Agents register their services and IDs
- Search interface : service → ID

Autonomous agents

Sender agent…

- Can send a message anytime
- Can't know if and when his message is being processed

Autonomous agents

Sender agent…

- Can send a message anytime
- Can't know if and when his message is being processed

Recipient agents…

- Can receive a message anytime
- Can process this message anytime

  Especially if they are overloaded…
- Can ignore a message

Autonomous agents

Sender agent…

- ▶ Can send a message anytime
- ▶ Can't know if and when his message is being processed

Recipient agents…

- ▶ Can receive a message anytime
- ▶ Can process this message anytime

  Especially if they are overloaded…
- ▶ Can ignore a message

Autonomy

Agents must take into account the autonomy of each other.

- ▶ But they cannot do anything ! → Protocols !

Interaction protocols I

Protocols

Describes how agents can interact in the MAS

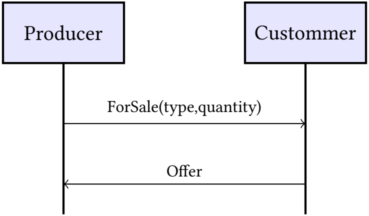
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML standard

- Inspired from UML sequence diagrams
- Describes message exchange between roles
  - An agent can adopt several roles
  - A role can be fulfilled by several different agents



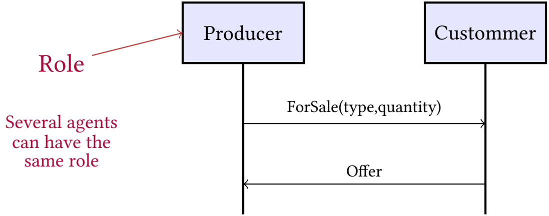
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML standard

- Inspired from UML sequence diagrams
- Describes message exchange between roles
  - An agent can adopt several roles
  - A role can be fulfilled by several different agents



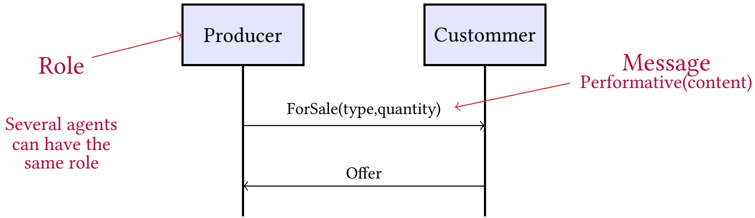
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML standard

- Inspired from UML sequence diagrams
- Describes message exchange between roles
  - An agent can adopt several roles
  - A role can be fulfilled by several different agents



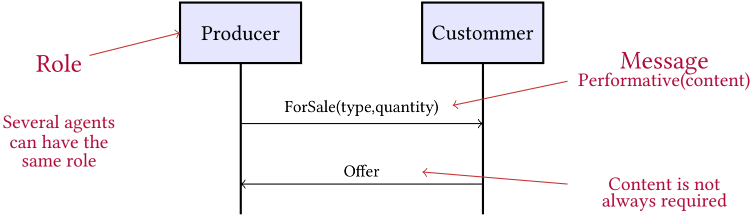
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML standard

- Inspired from UML sequence diagrams
- Describes message exchange between roles
  - An agent can adopt several roles
  - A role can be fulfilled by several different agents



Producer

Custommer

ForSale(type,quantity)

Offer

Role

Several agents can have the same role

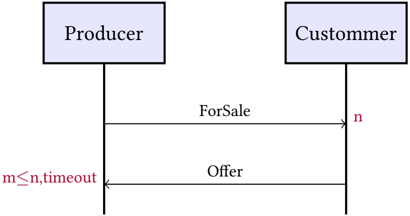
Message
Performative(content)

Content is not always required

Interaction protocols II

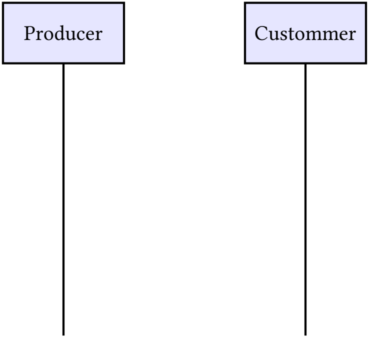
Conditions

- Number of messages sent (arrow end)
- Timeouts
  - Messages received after timeout are considered out of the protocol



Interaction protocols III

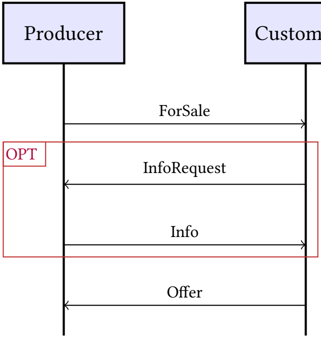
Operators



Interaction protocols III

Operators

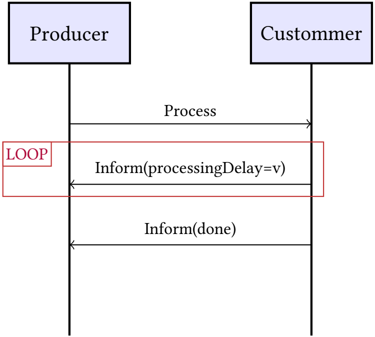
- OPT → some parts can be are optional



Interaction protocols III

Operators

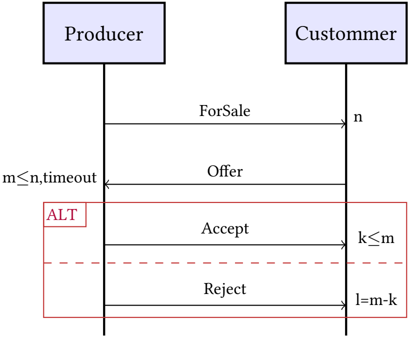
- OPT → some parts can be are optional
- LOOP → some parts can be repeted randomly



Interaction protocols III

Operators

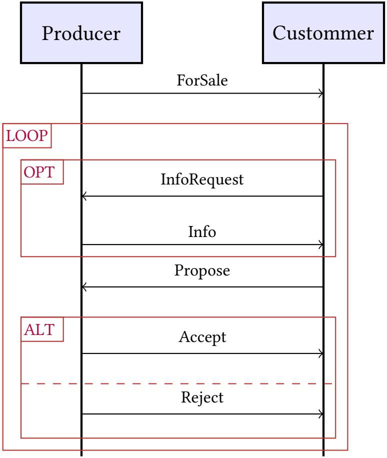
- OPT → some parts can be are optional
- LOOP → some parts can be repeted randomly
- ALT → one or the other



Interaction protocols IV

Operators

- Operators can be **combined**

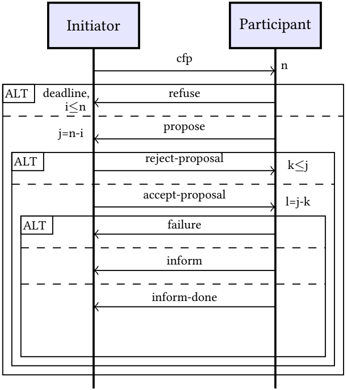


Interaction protocols V

Contract-Net Protocol

Standard for agents to agree on a transaction.

- FIPA standard
- The « must-know » protocol



What's next ?

Tutorial

Using the JADE platform for MAS programming