

Agent and Multi-Agent Systems: architectures and reasoning

Interaction mechanisms : models and implementation

Wassila Ouerdane

28.09.2021

CentraleSupélec- SAFRAN AI Training

Table of contents

1. Interaction Mechanisms
2. Indirect interactions
3. Direct interactions
4. Interaction protocols
5. Conclusion

Interaction Mechanisms

Interaction

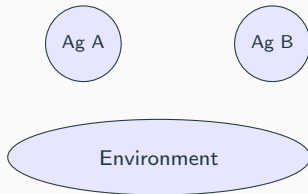
- An interaction occurs when two or more agents are brought into a dynamic relationship through a set of reciprocal actions.
- Interactions develop out of a series of actions whose consequences in turns have an influence on the future behavior of agents.

Problems in MAS

- Agents run asynchronously
- Method invocation is synchronous

PRS Architecture

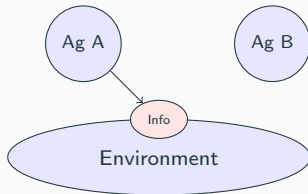
- Actions modify the environment
- (Asynchronous) perception of the modification



Asynchronous interactions

PRS Architecture

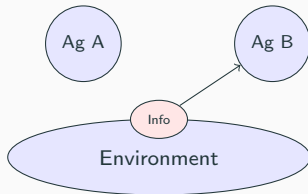
- Actions modify the environment
- (Asynchronous) perception of the modification



Asynchronous interactions

PRS Architecture

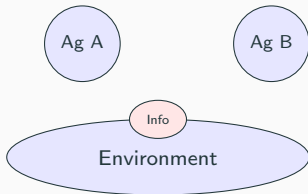
- Actions modify the environment
- (Asynchronous) perception of the modification



Asynchronous interactions

PRS Architecture

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



Interaction situations

- According to compatibility of **goals**
 - Agents **cooperate** when their goals are compatible
 - Agent **compete** when their goals are incompatible
- According to agent ability to available **resources**
 - **Conflicts** arises when resources are insufficient.
- According to agent ability to fulfill **tasks**
 - **Collaboration** arises when agent have insufficient ability to solve complex problems

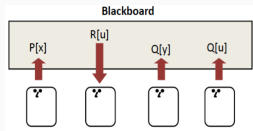
Indirect interactions

Indirect interactions

- No intention to communicate to a specific agent
- Agents interact through an intermediate entity
- This medium supplies specific interaction mechanisms and access rules
- These rules and mechanisms define agent local context and perception

Artifact-mediated interaction: **Blackboard systems**

- Agents access a shared artifact (stores data) that: they can observe and can modify
- Such artifact is a **communication channel** characterized by an intrinsically broadcast transmission.
- Specific **laws regulating access** to this medium
- It represents a part of agents' **environment**.



- The PLAN component of the Mission Control System for RADARSAT-1: an Earth observation satellite developed by Canada to monitor environmental changes and Earth's natural resources.
- Adobe Acrobat Capture : to decompose and recognize image pages to understand the objects, text, and fonts on the page (Adobe Acrobat as "OCR Text Recognition").
- Blackboard is used routinely in many military C4ISTAR systems for detecting and tracking objects.

Direct interactions

Principle

- **Intention** to communicate to a **specific agent**
 - Messages with sender, recipient and structured content
- Information exchange
 - Communication/Conversation rules (“protocols”): **Agent Communication Language (ACL)**
 - Message structure (shared ontology): **Content Language**

With whom?

With which agent to interact to obtain a service, a resource, ...?

Problem: heterogeneity

Communication layers:

- Transport level → environment (network)
- Syntax level → message structure
- Semantic level → knowledge representation
- Pragmatic (communication) level → protocols

⇒ This has to be normalised!

Searle, 1969

Communication is an action

Communicate → change interlocutor's mental state

Searle, 1969

Communication is an action

Communicate → change interlocutor's mental state

Three aspect of a speech act

- **Locutionary**: the act of saying (What was said. "Is there any salt?")
- **Illocutionary**: the intention of the speech (What was done. "Please give me some salt")
- **Perlocutionary**: the effect of the speech (What happened as a result. "might cause somebody to pass the salt")

Searle, 1969

Communication is an action

Communicate → change interlocutor's mental state

Three aspect of a speech act

- **Locutionary**: the act of saying (What was said. "Is there any salt?")
- **Illocutionary**: the intention of the speech (What was done. "Please give me some salt")
- **Perlocutionary**: the effect of the speech (What happened as a result. "might cause somebody to pass the salt")

Illocutionary act

Performative Verb (Propositional Content)

Examples

Content = 'the door is closed'

- Performative = request
 - speech act= 'please close the door'
- Performative = inform
 - speech act= 'the door is closed'
- Performative = inquire
 - speech act= 'is the door closed?'

Different researchers, different theories. . .

Searle 1972

- Assertive /representatives acts: *Facts*(informing, making a claim)
~> 'it is raining'
- Directive acts: *Actions + Questions* (requesting, commanding)
~> 'please make the tea'
- Promissive /Commissive acts: *Commitments*(promising , refusing)
~> 'I promise to ...'
- Expressive acts: *Emotions* (expressing mental states)
~> 'thank you!'
- Declarative acts: *Protocols* (effecting change to state of the world)
~> 'such as declaring war or naming'

Different researchers, different theories. . .

Searle 1972

- Assertive /representatives acts: *Facts* (informing, making a claim)
- Directive acts: *Actions + Questions* (requesting, commanding)
- Promissive /Commissive acts: *Commitments*(promising , refusing)
- Expressive acts: *Emotions* (expressing mental states
- Declarative acts: *Protocols* (effecting change to state of the world)

Sperber & Wilson

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

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

Agent Communication Languages

- **Agent communication languages (ACLs)** define standards for messages exchanged among agents
- Usually based on speech act theory, messages are specified by:
 - Sender/ receiver(s) of the message
 - Performative to describe intended actions
 - Propositional content in some content language
- Most commonly used languages:
 - KQML-KIF
 - FIPA- ACL (today the de-facto standard)

FIPA: Foundation for Intelligent Physical Agents

KQML: Knowledge Query and Manipulation Language

KIF: Knowledge Interchange Format

KQML- KIF: message format

```
(performative
  :sender      <word>      :receiver    <word>
  :in-reply-to <word>      :reply-with  <word>
  :language    <word>      :ontology     <word>
  :content      <expression>
)
```

Remark : KQML and Ontology

- In order to be able to communicate, agents need to agree on the words (terms) they use to describe the domain.
- A formal specification of a set of terms is known as an [ontology](#)
- The role of an ontology is to fix the meaning of the terms used by agents.

BlocksWorld

Blocksworld Ontology

<i>On(x,y)</i>	<i>object x on top of object y</i>
<i>OnTable(x)</i>	<i>object x is on the table</i>
<i>Clear(x)</i>	<i>nothing is on top of object x</i>
<i>Holding(x)</i>	<i>arm is holding x</i>

KQML/KIF

- KQML/KIF were very successful, but also some problems
- List of performatives (up to 41!) not fixed (interoperability problems)
- No formal semantics, only informal descriptions of meaning
- KQML completely lacks commissives, this is a massive restriction!
- Performative set of KQML rather ad hoc, not theoretically clear or very elegant

⇒ These lead to the development of FIPA ACL

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 (20 performative in FIPA)
- Content: expressed in any **knowledge representation** language
 - *First Order Logics, Lisp syntax, ...*

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 (20 performative in FIPA)
- Content: expressed in any **knowledge representation** language
 - *First Order Logics, Lisp syntax, ...*

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

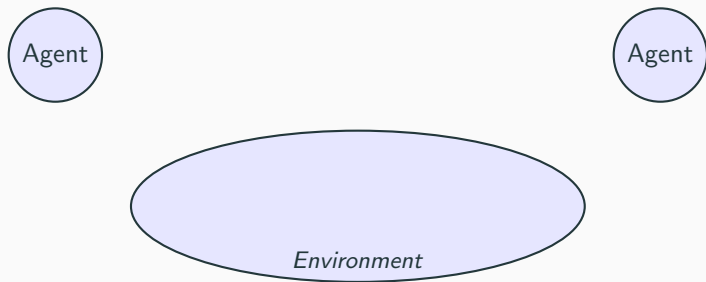
Example

```
(inform
  : sender agent 1
  : receiver agent 5
  : content (price good200 150)
  : language sl
  : ontology hpl-auction
)
```

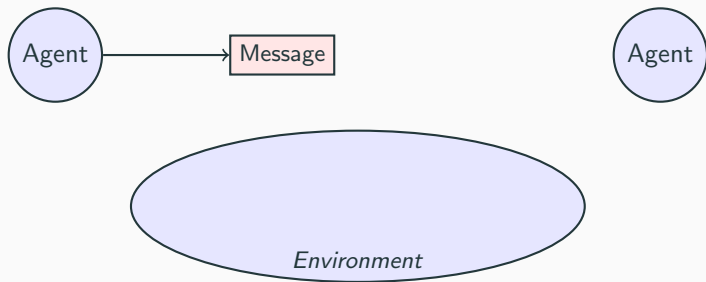
FIPA ACL Performatives

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

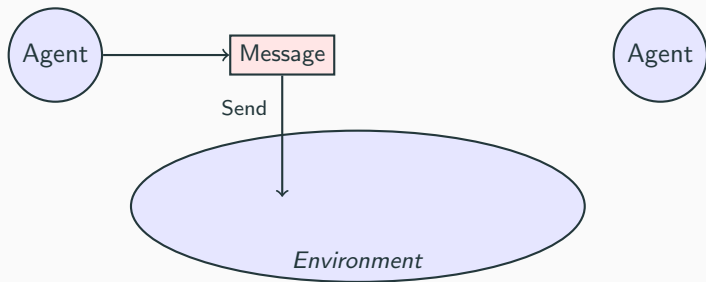
How does this work? Asynchronous message sending



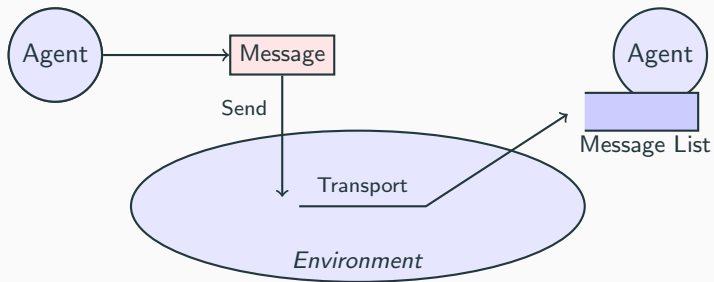
How does this work? Asynchronous message sending



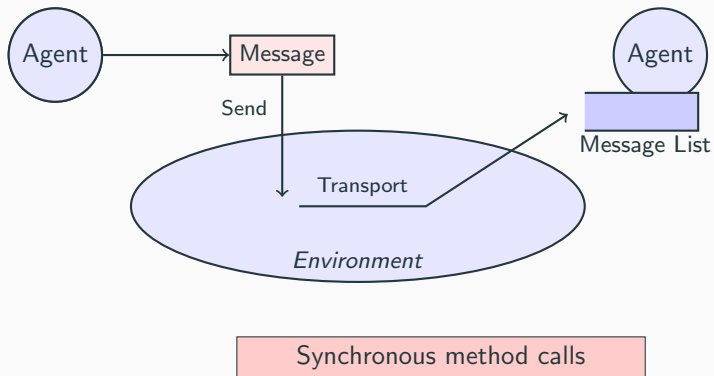
How does this work? Asynchronous message sending



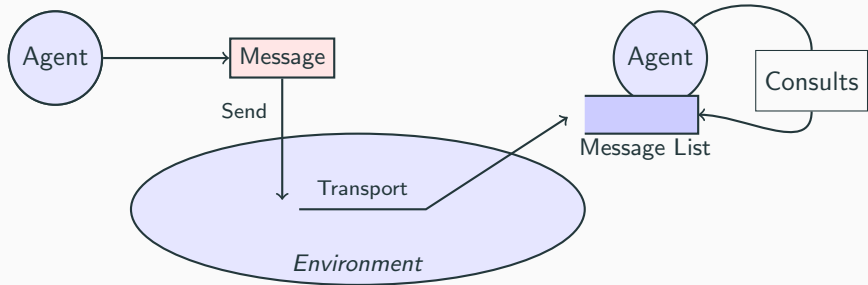
How does this work? Asynchronous message sending



How does this work? Asynchronous message sending



How does this work? Asynchronous message sending



Synchronous method calls

Asynchronous

Interaction protocols

Protocols

Describes how agents can interact in the MAS

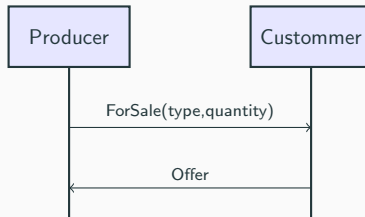
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML (Agent Unified Modeling language) 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



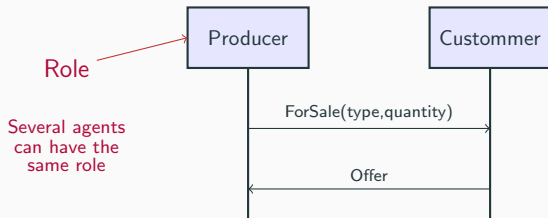
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML (Agent Unified Modeling language) 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



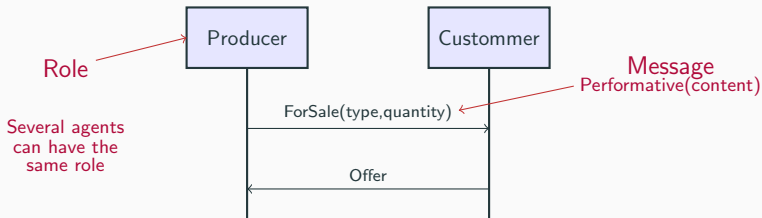
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML (Agent Unified Modeling language) 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



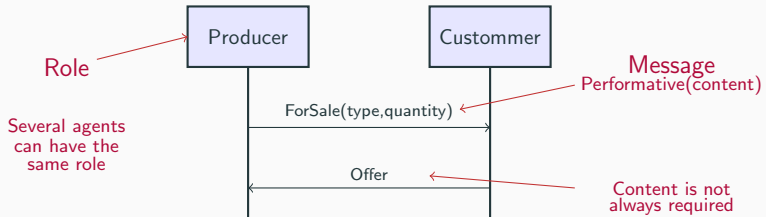
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML (Agent Unified Modeling language) standard

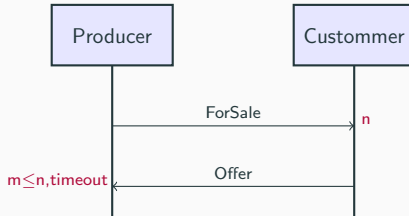
- Inspired from UML sequence diagrams
- Describes message exchange between **roles**
 - An agent can adopt several roles
 - A role can be fulfilled by several different agents



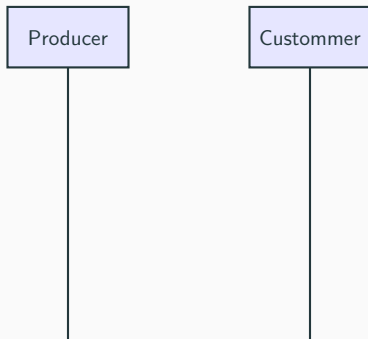
Interaction protocols II

Conditions

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



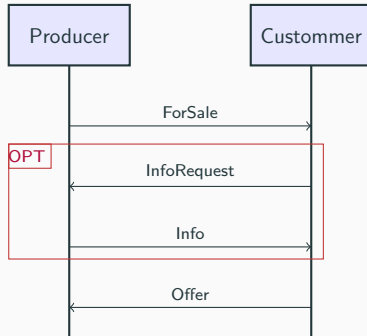
Operators



Interaction protocols III

Operators

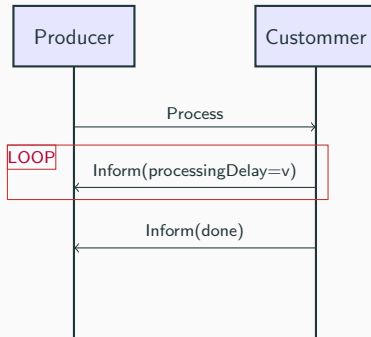
- OPT → some parts can be optional



Interaction protocols III

Operators

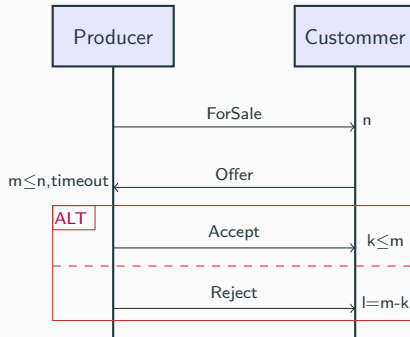
- OPT → some parts can be optional
- LOOP → some parts can be repeated randoml



Interaction protocols III

Operators

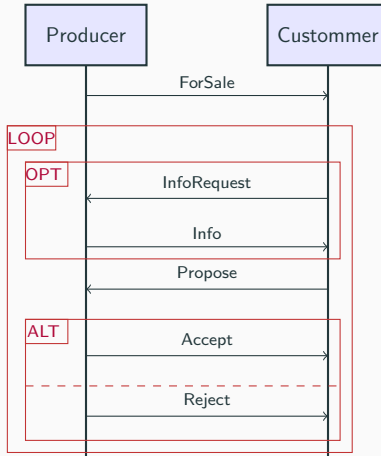
- OPT → some parts can be optional
- LOOP → some parts can be repeated randomly
- ALT → one or the other



Interaction protocols IV

Operators

- Operators can be **combined**

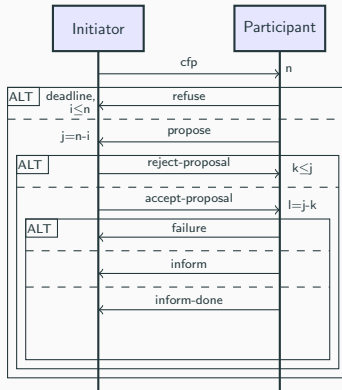


Interaction protocols V

Contract-Net Protocol (CNP)

Standard for agents to **agree on a transaction**.

- FIPA standard
- The “must-know” protocol



Contract-Net Protocol (CNP)

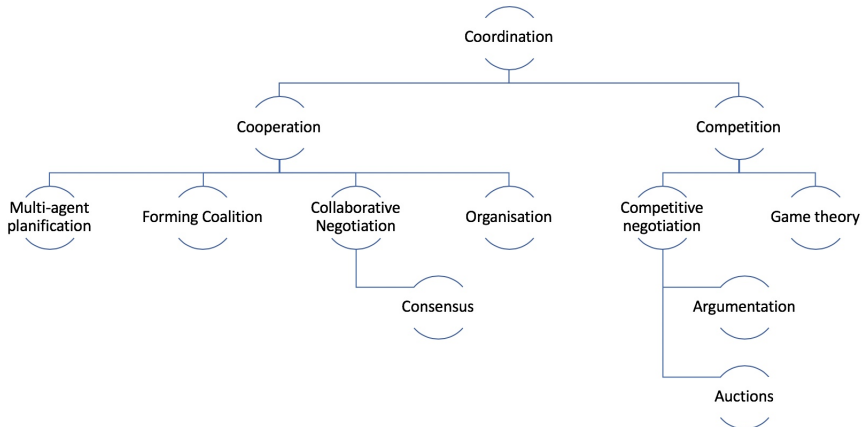
- One of the oldest, most widely used agent interaction protocols
- A manager agent announces one or several tasks, agents place bids for performing them
- Task is assigned by manager according to evaluation function applied to agents' bids (e.g., choose cheapest agent)
- Idea of exploiting local cost function (agents' private knowledge) for distributed optimal task allocation
- Even in purely cooperative settings, decentralization can improve global performance
- Successfully applied to different domains (e.g. transport logistics)

Conclusion

To resume

- Different kinds of interaction and communication
- Focus on agent-to-agent communication
- Speech act theory - theoretical foundation for ACL
- Agent communication languages & their semantics
- Interaction protocols
- But how about agent strategies in interaction and their global effects? Next time: [Methods for Coordination/Negotiation](#)

Coordination Mechanisms



Source: borrowed from Aurélie Beynier's course