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

Interaction Mechanisms

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.

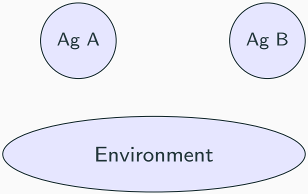
Problems in MAS

- Agents run asynchronously
- Method invocation is synchronous

Asynchronous interactions

PRS Architecture

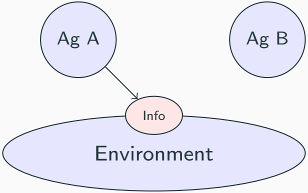
- Actions modify the environment
- (Asynchronous) perception of the modification



Asynchronous interactions

PRS Architecture

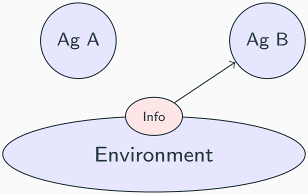
- Actions modify the environment
- (Asynchronous) perception of the modification



Asynchronous interactions

PRS Architecture

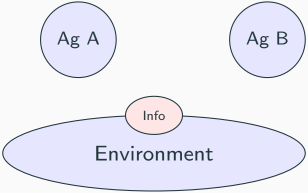
- Actions modify the environment
- (Asynchronous) perception of the modification



Asynchronous interactions

PRS Architecture

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



Interaction Mechanisms

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

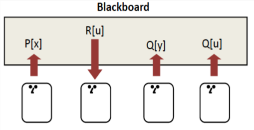
Indirect interactions

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.



For more details
https://en.wikipedia.org/wiki/Blackboard_system

Direct interactions

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

Direct interactions

With whom?

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

Problem: heterogeneity

Communication layers:

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

$\Rightarrow$ This has to be normalised!

Speech Acts Theory I

Searle, 1969

Communication is an action

Communicate $\rightarrow$ change interlocutor's mental state

Speech Acts Theory I

Searle, 1969

Communication is an action

Communicate $\rightarrow$ 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")

Speech Acts Theory I

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)

Speech Acts Theory I

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?'

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

Message structure

KQML- KIF: message format

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

Message structure

Remark : KQML and Ontology

- In oder 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

On(x,y)	*object x on top of object y*
OnTable(x)	*object x is on the table*
Clear(x)	*nothing is on top of object x*
Holding(x)	*arm is holding x*

Message structure

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

$\leadsto$These lead to the development of FIPA ACL

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

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

FIPA-ACL standard

Example

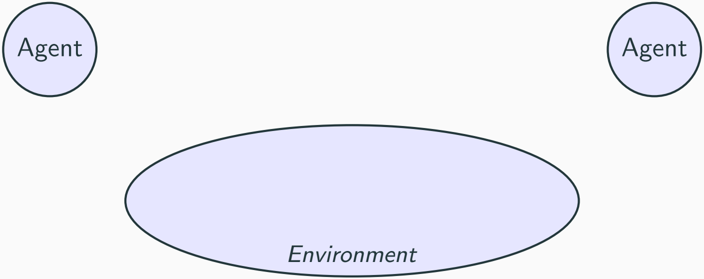
```
(inform
  : sender agent 1
  : receiver agent 5
  : content (price good200 150)
  : language sl
  : ontology hpl-auction
 )
```

FIPA-ACL standard

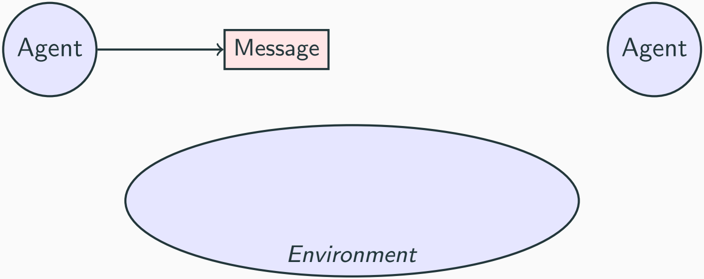
FIPA ACL Performatives

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			

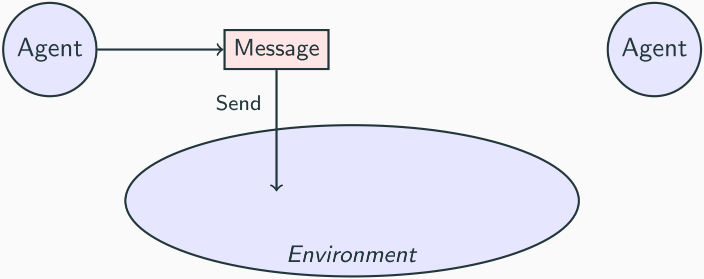
How does this work? Asynchronous message sending



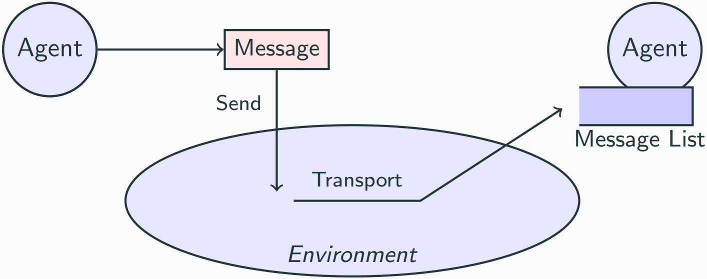
How does this work? Asynchronous message sending



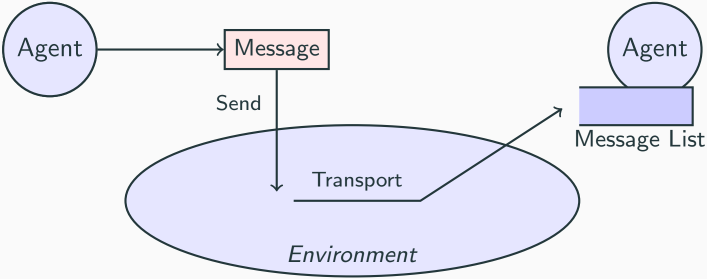
How does this work? Asynchronous message sending



How does this work? Asynchronous message sending

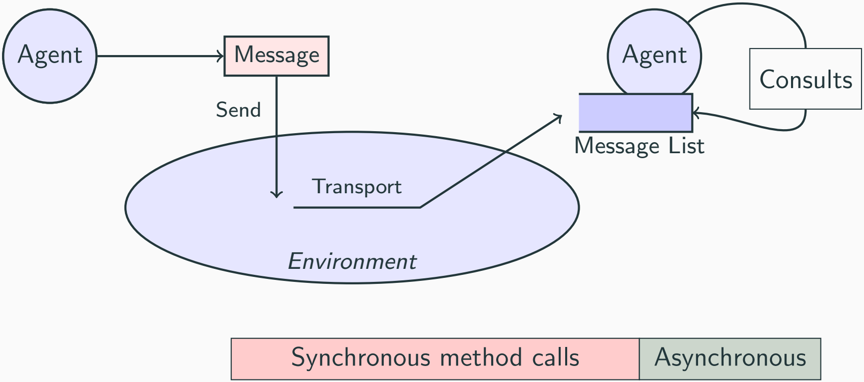


How does this work? Asynchronous message sending



Synchronous method calls

How does this work? Asynchronous message sending



Interaction protocols

Interaction protocols I

Protocols

Describes how agents can interact in the MAS

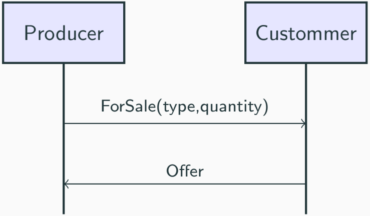
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML (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



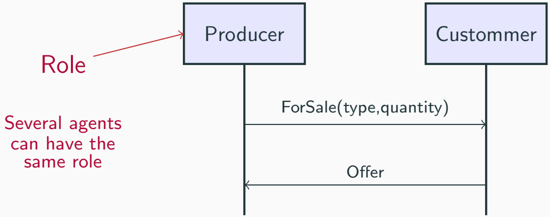
Interaction protocols I

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



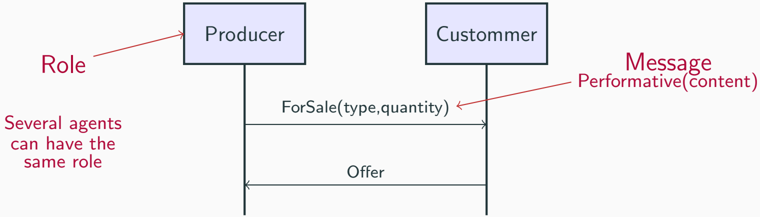
Interaction protocols I

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



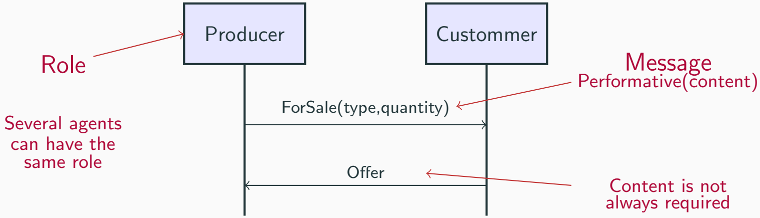
Interaction protocols I

Protocols

Describes how agents can interact in the MAS

AUML (Agent Unified Modeling language) standard

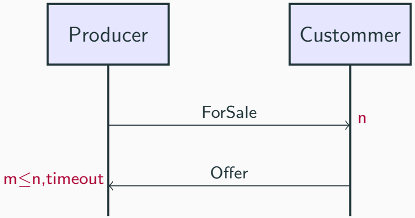
- Inspired from UML sequence diagrams
- Describes message exchange between roles
  - An agent can adopt several roles
  - A role can be fulfilled by several different agents



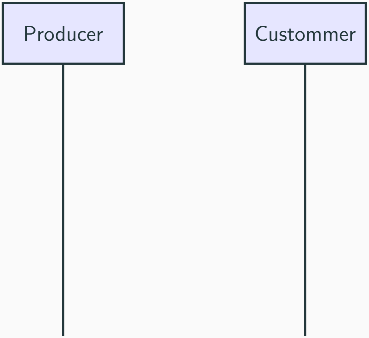
Interaction protocols II

Conditions

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



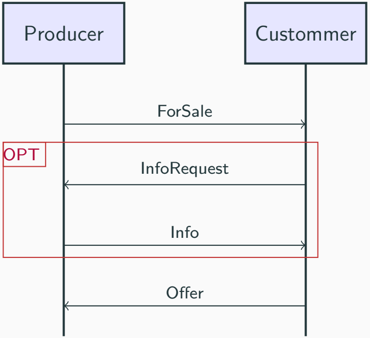
Operators



Interaction protocols III

Operators

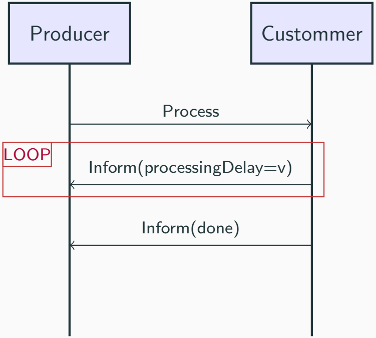
- OPT $\rightarrow$ some parts can be optional



Interaction protocols III

Operators

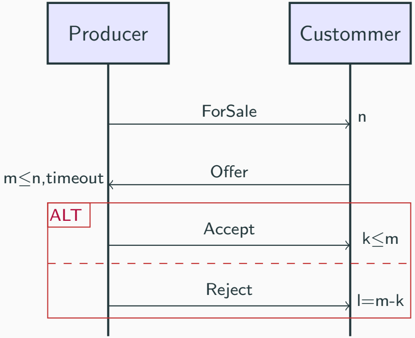
- OPT → some parts can be optional
- LOOP → some parts can be repeated randoml



Interaction protocols III

Operators

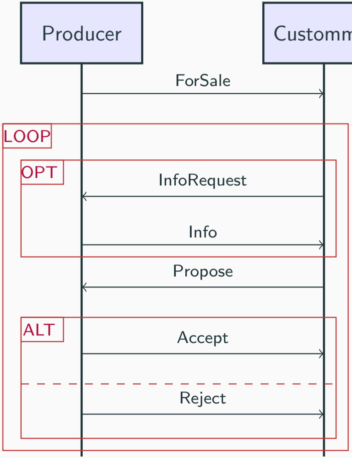
- OPT $\rightarrow$ some parts can be optional
- LOOP $\rightarrow$ some parts can be repeated randoml
- ALT $\rightarrow$ one or the other



Interaction protocols IV

Operators

- Operators can be combined



Interaction protocols

Contract-Net Protocol (CNP)

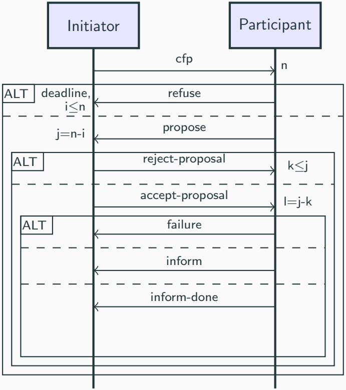
- One of the oldest, most widely used agent interaction protocols
- A manager agent announces one or several tasks, agents place bids for performing them
- Task is assigned by manager according to evaluation function applied to agents' bids (e.g., choose cheapest agent)
- Idea of exploiting local cost 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)

Interaction protocols V

Contract-Net Protocol (CNP)

Standard for agents to agree on a transaction.

- FIPA standard
- The “must-know” protocol



Dialogue Games

Dialogue Games

- Considerable efforts has been expended on the design of artificial languages for agent communication (KQML, FIPA ACL).
- However, agents participating in conversations have too many choices of what utter at each turn $\rightsquigarrow$ agent dialogues may endure a state-space explosion.
- Appearance of Dialogue games: rule-governed interactions between two or more players (or agents), where each player “movers” by making utterances, according to defined set of rules.

Dialogue types [Walton and Krabbe, 1995]

A categorization based upon the information the participants have at the commencement of a dialogue, their individual goals for the dialogue and the goals they share.

Types of dialogue	Initial situation	Participant's goal	Goal of dialogue
Persuasion	Conflicts of opinions	Persuade other party	Resolve or clarify issue
Negotiation	Conflicts of interests	Get what you most want	Reasonable settlement
Information seeking	One party lacks information	Acquire or give information	Exchange Information
Deliberation	Dilemma or practical choice	Coordinate goals or actions	Decide best course of action
...			

A negotiation dialogue

Example

Buyer :	Can't you give me this 806 a bit cheaper?
Seller :	Sorry that's the best I can do. Why don't you go for a Polo instead?
Buyer :	I have a big family and I need a big car (B1)
Seller :	Modern Polo are becoming very spacious and would easily fit in a big family. (S1)
Buyer :	I didn't know that, let's also look at Polo then

Approaches to automated negotiation

- Negotiation is essential in settings where autonomous agents have conflicting interests and a desire to cooperate
- Various interaction and decision mechanisms for automated negotiation have been proposed and studied:

Game-theoretic analysis (Rosenschein & Zlotkin, 1994; Kraus, 2001; Sandholm, 2002);

- Aim: to determine the optimal strategy by analyzing the interaction as a game between identical participant and seeking its equilibrium.

✗ assumes that agent have unbounded computational resources and that the space of outcomes is completely known.

✗ Agent's utilities or preferences are usually assumed to be completely characterized prior to interaction ($\leadsto$incomplete information?), and fixed (no possibility to influence another agent's preference model or internal mental attitudes: beliefs, desires goals, etc.)

Approaches to automated negotiation

- Negotiation is essential in settings where autonomous agents have conflicting interests and a desire to cooperate
- Various interaction and decision mechanisms for automated negotiation have been proposed and studied:

Heuristic-based approaches (Faratin, 2000; Kowalczyk & Bui, 2001; Fatima et al., 2002):

- Aim: use heuristics rules that produce good enough, rather than optimal, outcomes/decisions.
- ✗ Outcomes are sub-optimal (do not examine the full space of possible outcomes)
- ✗ It is difficult to predict precisely how the system and the constituent agents will behave, thus the models need extensive evaluation through simulation and empirical analysis.
- ✗ Agent's utilities or preferences are usually assumed to be completely characterized prior to interaction ($\rightsquigarrow$incomplete information?), and fixed (no possibility to influence another agent's preference model or internal mental attitudes: beliefs, desires goals, etc.)

Approaches to automated negotiation

Agents might:

- lack some information relevant to making a comparison between two potential outcomes,
- have limited resources preventing them from acquiring such information
- have the information, but lack the time needed to process it in order to make the comparison or the choice,
- have incoherent preferences,
- ...

$\rightsquigarrow$ process of acquiring information, resolving uncertainties, revising preferences, etc often take places as part of the negotiation itself.

Approaches to automated negotiation

- Negotiation is essential in settings where autonomous agents have conflicting interests and a desire to cooperate
- Various interaction and decision mechanisms for automated negotiation have been proposed and studied:

Argumentation-based approaches (Kraus et al., 1998; Parsons et al., 1998; Sierra et al., 1998)

- allow to exchange different kinds of information, to "argue" about their beliefs and other mental attitudes during the negotiation process.
- an argument can be viewed as a piece of information to i) *justify* its negotiation stance, or ii) *influence* another agent's negotiation stance

Argumentation

- Argumentation is the process of attempting to agree about what to believe or what to do.
- Only a question when information, beliefs, actions, ...are contradictory
  If everything is consistent, just merge information from multiple agents.
- Argumentation provides principled techniques for resolving inconsistency (conflicts). Or at least, sensible rules for deciding what to believe or what to do in the face of inconsistency.

Where argumentation appear?

We find argumentation in:

- Consumer websites: Amazon, eBay, . . . ;
- Law: policy making, Supreme transcripts, case based reasoning, regulations;
- Medical diagnosis;
- Making plans;
- Debatepedia, Wikipedia, meeting annotation, web-forums, . . . ;
- Social media: Facebook, dating, etc.
- . . .

Defeasible Reasoning

- Reasoning is generally defeasible [1] (assumptions, exceptions, uncertainty, ...)
- AI formalises such reasoning with non-monotonic logics (e.g. Default logic, etc.):
  $\rightsquigarrow$new premisses can invalidate old conclusions
- Argumentation logics formalise defeasible reasoning as construction and comparison of arguments

[1] révocable

Logic

Logic

- Accepted rules for making precise statements.
- Logic for computer science: programming, artificial intelligence, logic circuit, database
- Logic
    - Represents knowledge precisely
    - Helps to extract information (inference)

Propositional Logic

- Proposition: a statement that is either true or false but not both.
- Uses logical operator to combine statements: $\wedge$ (conjunction, and), $\vee$ (disjunction, or), $\neg$ (negation, not), $\rightarrow$ (conditional statement,if p then q)
- inference rule: is a logical construct which tales premises, analyses syntax and returns conclusion. e.g. Modus ponens rule:

$$A \rightarrow B \, , A \vdash B$$

(A implies B, and A is asserted to be true, therefore B must true)

- $\vdash$: classical consequence relation (entailment)

Non-monotonic reasoning

- Often available knowledge is incomplete. However, to model commonsense reasoning, it is necessary to be able to jump to plausible conclusions from the given knowledge.
- To draw plausible conclusions it is necessary to make assumptions.
- The choice of assumptions is not blind: most of the knowledge on the world is given by means of general rules which specify typical properties of objects.

Non-monotonic reasoning

- Non-monotonic reasoning deals with the problem of deriving plausible conclusions, but not infallible, from a knowledge base (a set of formulas).
- Since the conclusions are not certain, it must be possible to retract some of them if new information shows that they are wrong
- Classical logic is inadequate since it is monotonic: if a formula B is derivable from a set of formulas S, then B is also derivable from any superset of S:

$$S \vdash B \text{ implies } S \cup \{A\} \text{ for any formula } A$$

Non-monotonic reasoning?

Example (Reiter,1987)

- Birds fly;
- Tweety is a bird;
- Therfore, Tweety flies.

But what if Tweety= penguin, type of bird that does not fly?

Non-monotonic reasoning?

Example (Reiter,1987)

- Birds fly;
- Tweety is a bird;
- Tweety is a penguin;
- ~~Therfore, Tweety flies.~~ False

Non-monotonic reasoning?

Example (Reiter,1987)

- Birds fly;
- Tweety is a bird;
- Tweety is a penguin;
- ~~Therfore, Tweety flies.~~ False

Consequence (classical Logic)

The “Birds fly” must be false !!!

Non-monotonic reasoning?

We can say

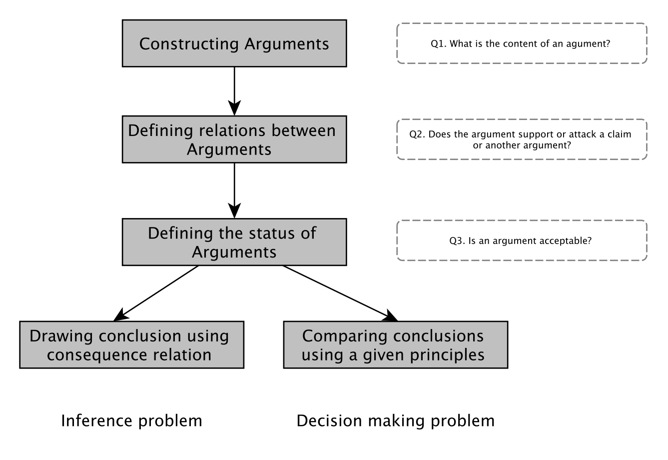
- "Normally, birds fly" or "if x is a typical bird, then we can assume by default that x flies";
- default $\rightsquigarrow$ in the absence of evidence that Tweety is atypical, we can provisionally assume that Tweety flies.

Non-monotonic reasoning: an example

Legal reasoning

- most legal regulations are rules with exceptions
- different legal regulations can hold at the same time. Which overwrites which?
  E.g.: state law can overwrite local law
- defaults can represent this naturally

A standard argumentation process



Step 1: constructing arguments?

Case 1: Inference Problem

- Let Σ be a propositional knowledge base
- An argument A is a pair $A = (\Phi, \alpha)$, such that
  1. $\Phi \subset \Sigma$
  2. Φ is consistent [2]
  3. $\Phi \vdash \alpha$
  4. Φ is minimal (for set inclusion) satisfying 1, 2 and 3.

  Φ: premises (supports) and α: conclusion (consequence).

[2] if there is no formula ϕ such that ϕ such that $\Phi \vdash \phi$ and $\Phi \vdash \neg\phi$

Step 1: constructing arguments?

Exemple [Cayrol et al. 2005]

A murder has been performed and the suspects are Liz, Mary and Peter. The following pieces of information have been gathered:

- The type of murder suggests us that the killer is a female;
- The killer is certainly small;
- Liz is tall and Mary and Peter are small.
- The killer has long hair and uses a lipstick;
- A witness claims that he saw the killer who was tall;
- We are told that the witness is short-sighted, so he is no more reliable.

$\rightarrow$ *Who is the killer?*

Step 1: constructing arguments?

Exemple [Cayrol et al. 2005] cont.

- s (the killer is small);
- f (the killer is a female);
- m (the killer is Mary);
- l (the killer has long hair and uses a lipstick);
- w (the witness is reliable);
- b (the witness is short-sighted).

Arguments

- A_1: $< \{s, f, s \wedge f \rightarrow m\}, m >$ (in favour of m);
- A_2: $< \{w, w \rightarrow \neg s\}, \neg s >$ (in favour of $\neg s$);
- A_3: $< \{b, b \rightarrow \neg w\}, \neg w >$ (in favour of $\neg w$);
- A_4: $< \{l, l \rightarrow f\}, f >$ (in favour of f).

Step 1: constructing arguments?

Case 2: Decision problem

- Let Σ be a propositional knowledge base
- G: a goals base
- D: a set of decision options
- An argument A in favor of a decision d is a triple $A = (S, g, d)$, such that
  - $d \in D$
  - $g \in G$
  - $S \subset \Sigma$
  - $S \cup \{d\}$ is consistent
  - $S \cup \{d\} \vdash g$
  - S is minimal (for set inclusion) satisfying the above conditions

Step 2: Interaction between arguments

Three conflicting relations

- Rebutting attacks: two argument with contradictory conclusions;
- Assumption attacks: an argument attacks an assumption of another argument;
- Undercutting attacks: an argument undermines some intermediate step (inference rule) of another argument.

Step 2: Interaction between arguments

Example

Argument x

Here is one deductive argument.

a denotes "*We recycle*"
b denotes "*We save resources*"
$a \rightarrow b$ denotes "*If we recycle, then we save resources*"

Formally we get: $(\{a, a \rightarrow b\}, b)$

Argument y

A second argument, that conflicts with the first:

c denotes "*Recycled products are not used*"
$a \wedge c \rightarrow \neg b$ denotes "*If we recycle and recycled products are not used then we don't save resources*"

Formally we get: $(\{a, c, a \wedge c \rightarrow \neg b\}, \neg b)$

x and y rebut each other.

Argument z

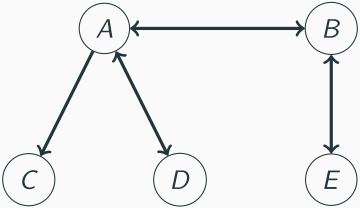
A third argument, that conflicts with the first:

d denotes "*We create more desirable recycled products*"
$d \rightarrow \neg c$ denotes "*If we create more desirable recycled products then recycled products are used*"

Formally we get: $(\{d, d \rightarrow \neg c\}, \neg c)$

z undercuts y

Abstract Argumentation system



Definition

An argumentation system is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$, where: $\mathcal{A}$ is a set of arguments and $\mathcal{R} \subset \mathcal{A} \times \mathcal{A}$: an attack relation among arguments.

Step 3 : Status of arguments?

- An argumentation frameworks encodes, through the attack relations, the existing conflicts within a set of arguments.
- A formal method to identify conflicts outcomes (i.e. determine which arguments should be accepted) for any argumentation framework is called argumentation semantics.
- Two mains approaches to the definition of argumentation semantics:
  - The labelling-based approach
  - The extension-based approach

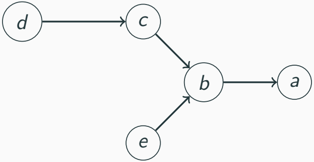
Extension-based approach [Dung, 1995]

Let an argumentation system $\langle \mathcal{A}, \mathcal{R} \rangle$, where: $\mathcal{A}$ is a set of arguments and $\mathcal{R} \subset \mathcal{A} \times \mathcal{A}$: an attack relation among arguments.

Definitions

Let $\mathcal{B} \subset \mathcal{A}$

- $\mathcal{B}$ is conflict-free iff $\nexists a, b \in \mathcal{B}$ such that $(a, b) \in \mathcal{R}$;
- $\mathcal{B}$ defends an argument a iff $\forall b \in \mathcal{A}$, if $(b, a) \in \mathcal{R}$, then $\exists c \in \mathcal{B}$ such that $(c, b) \in \mathcal{R}$



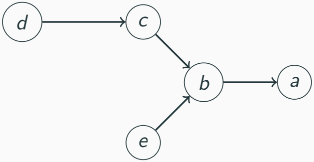
Extension-based approach [Dung, 1995]

Let an argumentation system $\langle \mathcal{A}, \mathcal{R} \rangle$, where: $\mathcal{A}$ is a set of arguments and $\mathcal{R} \subset \mathcal{A} \times \mathcal{A}$: an attack relation among arguments.

Definitions

Let $\mathcal{B} \subset \mathcal{A}$

- $\mathcal{B}$ is conflict-free iff $\nexists a, b \in \mathcal{B}$ such that $(a, b) \in \mathcal{R}$;
- $\mathcal{B}$ defends an argument a iff $\forall b \in \mathcal{A}$, if $(b, a) \in \mathcal{R}$, then $\exists c \in \mathcal{B}$ such that $(c, b) \in \mathcal{R}$



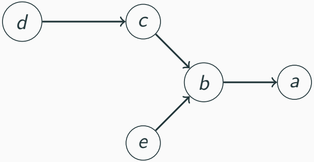
$\{a, e\}$ is conflict-free.

Extension-based approach [Dung, 1995]

Admissible extensions

Let $\mathcal{B} \subset \mathcal{A}$, $\mathcal{B}$ is an admissible extension iff:

- $\mathcal{B}$ is conflict-free;
- $\mathcal{B}$ defends all its elements.
- It is a minimal notion of a reasonable position (internally consistent and defends itslef, and it is coherent, defendable position).

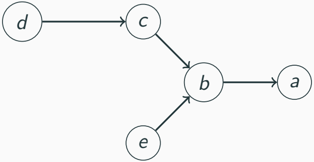


Extension-based approach [Dung, 1995]

Admissible extensions

Let $\mathcal{B} \subset \mathcal{A}$, $\mathcal{B}$ is an admissible extension iff:

- $\mathcal{B}$ is conflict-free;
- $\mathcal{B}$ defends all its elements.
- It is a minimal notion of a reasonable position (internally consistent and defends itslef, and it is coherent, defendable position).



$\{\}$,$\{d\}$,$\{e, a\}$, $\{d, e\}$,$\{d, e, a\}$

Extension-based approach [Dung, 1995]

Stable extensions

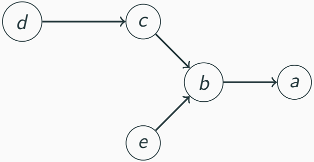
Let $\mathcal{B} \subset \mathcal{A}$, $\mathcal{B}$ is a stable extension iff:

- $\mathcal{B}$ is conflict-free;
- $\mathcal{B}$ attacks any argument in $\mathcal{A} \setminus \mathcal{B}$

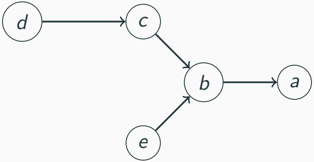
Notes

- intuition: un argument is not accepted because it is attacked by at least one accepted argument;
- it does not exist necessarily a stable extension, however we might have several stable extensions;

Extension-based approach [Dung, 1995]

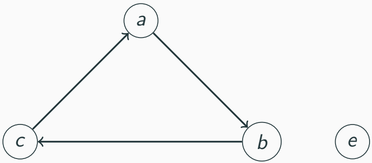


Extension-based approach [Dung, 1995]

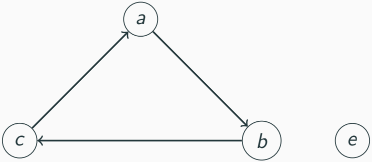


Stable: $\{d, e, a\}$

Extension-based approach [Dung, 1995]

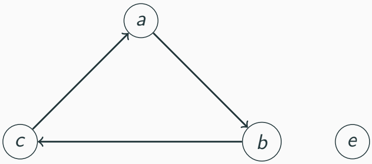


Extension-based approach [Dung, 1995]



- No stable extension $\rightarrow$ no accepted argument

Extension-based approach [Dung, 1995]



- No stable extension $\rightarrow$ no accepted argument
- But we would like to accept the argument e since it is not attacked !

Extension-based approach[Dung, 1995]

Preferred extensions

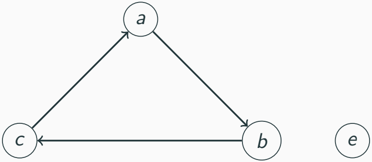
Let $\mathcal{B} \subset \mathcal{A}$, $\mathcal{B}$ is a preferred extension iff:

- $\mathcal{B}$ is an admissible extension;
- $\mathcal{B}$ is maximal for set inclusion among admissible extensions.

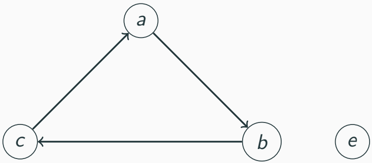
Notes

- intuition: it represents maximal coherent positions, able to defend themselves against all attackers.
- it necessarily exists a preferred extension (we can have several ones also)
- every stable extension is a preferred extension (the inverse is not true).

Extension-based approach [Dung, 1995]



Extension-based approach [Dung, 1995]



- One preferred extension $\rightarrow \{\mathrm{e}\}$
- e is accepted

Abstract argumentation system

$\rightsquigarrow$What is the status of an argument a in $\mathcal{A}$?

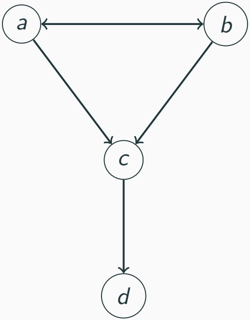
Abstract argumentation system

$\rightsquigarrow$What is the status of an argument a in $\mathcal{A}$?

Let $\mathcal{E}_1, \ldots, \mathcal{E}_k$: the extensions (under a given semantics) of $\langle \mathcal{A}, \mathcal{R} \rangle$

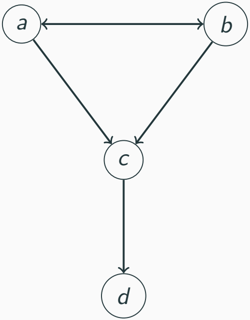
- a is justified iff $\forall \mathcal{E}_{i=1,\ldots,k}, a \in \mathcal{E}_i$
- a is defensible iff $\exists \mathcal{E}_i$ such that $a \in \mathcal{E}_i$
- a is rejected (overruled) $\nexists \mathcal{E}_i$ such that $a \in \mathcal{E}_i$

Abstract argumentation system



Preferred extensions $\rightsquigarrow \{a, d\}, \{b, d\}$. We can say:

Abstract argumentation system



Preferred extensions $\rightsquigarrow \{a, d\}, \{b, d\}$. We can say:

- d is justified
- c is overruled (rejected)
- a and b are defensible (undecided)

Step 4 : How to conclude?

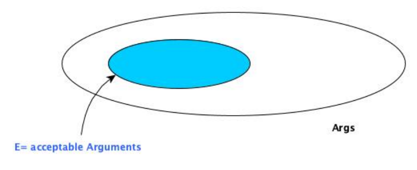
Making Decisions

- D a set of decision options
- Problem: to define a preordering on D
- $\langle \mathcal{A}, \mathcal{R} \rangle$: an argumentation system.

Let $d \in D$:

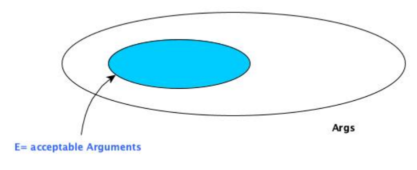
$$\langle \underbrace{P_1, \ldots, P_n}_{Arg.PRO\ d}, \underbrace{C_1, \ldots, C_m}_{Arg.CON\ d} \rangle$$

Step 4 : How to conclude?



- $\mathrm{Arg}_P(d)$: argument in E wich are PRO d
- $\mathrm{Arg}_C(d)$: argument in E wich are CON d

Step 4 : How to conclude?



- $\text{Arg}_P(d)$: argument in E wich are PRO d
- $\text{Arg}_C(d)$: argument in E wich are CON d

Decision principles: examples

- Unipolar principles: only one kind of arguments (PRO or CON) is involved (ex: counting the number of arguments)
- Bipolar principles: both argument PRO and CON are involved (cardinality, aggregation mechanisms / rules, value-based, etc.).

Argumentation-based negotiation

Abstractly, a negotiation framework can be viewed in terms of :

- negotiating agents: internal motivations, decision mechanisms, knowledge base, etc.
- an environment in which agents interact: rules of interaction, communication language, information store, etc.
  - Communication language + Domain Language
  - Protocol : the set of rules for generating coherent dialogues.

Environment and Components of a Negotiation Dialogue

Communication language

- A syntax: a set of locutions, utterances or speech acts (propose, argue, accept, reject, etc.) ($\rightsquigarrow$ e.g. FIPA-ACL)
- A semantic : a unique meaning for each utterance (a locution gives the same content a different meaning).

Environment and Components of a Negotiation Dialogue

Communication language

- A syntax: a set of locutions, utterances or speech acts (propose, argue, accept, reject, etc.) ($\rightsquigarrow$ e.g. FIPA-ACL)
- A semantic : a unique meaning for each utterance (a locution gives the same content a different meaning).

Domain language

Concepts about the environment, the different agents, the time, proposals, etc.

Example [Sierra et al., 1998]

- `offer` (a, b, price=200$ and Item=palm130, t_1): means the agent a proposes to agent b at time t_1 the sale of item palm130 for the price 200$.
- `reject` (b, a, price=200$ and item =palm130, t_2): means that agent b rejects such a proposal made by agent a.

Environment and Components of a Negotiation Dialogue

Negotiation protocol: a formal set of convention governing the interaction among participants. Main components:

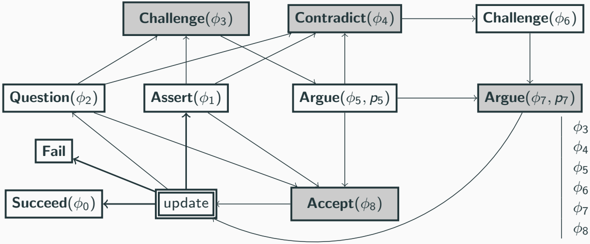
- interaction protocol: specifies, at each stage of the negotiation process, who is allowed to say what.
- some other rules:
  - rules for admission, which specify when an agent can participate in a negotiation dialogue and under what conditions;
  - termination rules, which specify when an encounter must end (e.g., if one agent utters an acceptance locution);
  - rules for proposal validity, which specify when a proposal is compliant with some conditions (e.g., an agent may not be allowed to make a proposal that has already been rejected);
  - rules for outcome determination, which specify the outcome of the interaction; in argumentation-based frameworks, these rules might enforce some outcome based on the underlying theory of argumentation (e.g., if an agent cannot construct an argument against a request, it accepts it [Parsons et al., 1998]);

Negotiation Protocol

How to specify an interaction protocol?

- Finite-state machines [Parsons et al., 1998]: explicit specification of interaction protocoles.
  $\rightsquigarrow$ Advantage: useful when the interaction involves a limited number of locutions
- Dialogue games [McBurney et al., 2003], by stating clearly the pre and post conditions of each locution as well as its effects on agent's commitments.
  $\rightsquigarrow$ Advantage: providing clear and precise semantics of the dialogue

Example I: finite-state machines



$\phi_3 \subseteq \phi_1$, $\phi_3 \subseteq \phi_2$
$\phi_4 \subseteq \phi_1$, $\phi_4 \subseteq \phi_2$, $\phi_4 \subseteq \phi_5$
$\phi_5 \subseteq \phi_3$
$\phi_6 \subseteq \phi_4$
$\phi_7 \subseteq \phi_6$, $\phi_7 \subseteq \phi_5$ or ϕ_7 conflicting with ϕ_5
$\phi_8 \subseteq \phi_1$, $\phi_8 \subseteq \phi_2$, $\phi_8 \subseteq \phi_5$

Ch. Labreuche, N. Maudet, W. Ouerdane, and S. Parsons. A dialogue game for recommendation with adaptive preference models. International Conference on Autonomous Agents and Multiagent systems, May 2015, Istanbul, Turkey. pp.959-967.

Example II: Dialogue games

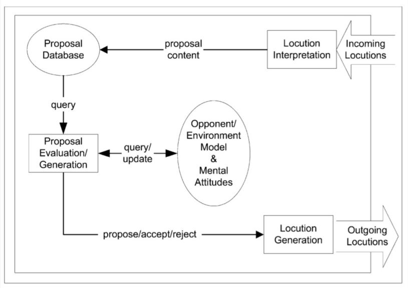
This locution allows a seller (or advisor) agent to announce that it (or another seller) is willing to sell a particular option [McBurney et al. ,2003].

- **Locution**: `willing_to_sell` (P_1, T, P_2, V), where P_1 is either an advisor or a seller, T is the set of participants, P_2 is a seller and V is a set of sales options.
- **Preconditions**: some participant P_3 must have previously uttered a locution `seek_info` (P_3, S, p) where $P_1 \in S$ (the set of sellers), and the options in V satisfy constraint p.
- **Meaning**: the speaker P_1 indicates to audience T that agent P_2 is willing to supply the finite set $V = \{a, b, \ldots\}$ of purchase options to any buyer in set T. Each of these options satisfy constraint p uttered as part of the prior `seek(.)` locution.
- Response: none required.
- **Information store updates**: for each $\neg a \in V$, the 3-tuple $(T, P_2, \neg a)$ is inserted into $IS(P_1)$, the information store for agent P_1.
- Commitment store updates: no effects.

Conceptual elements of a classical negotiating agent

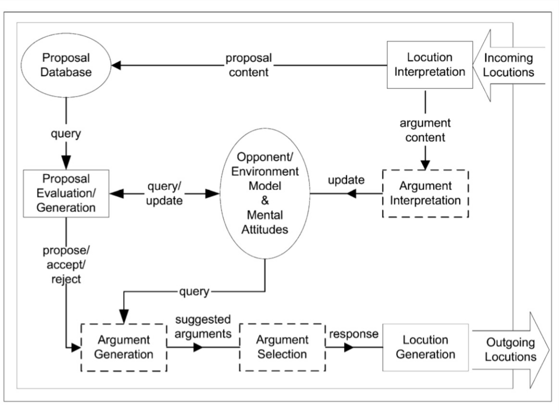
- Locution interpreter: parse incoming message
- Proposal evaluation and generation : makes a decision about whether to accept , reject, or generate a counterproposal, or terminate the negotiation.
- Locution generation: sends the response to the relevant party

Auction-based



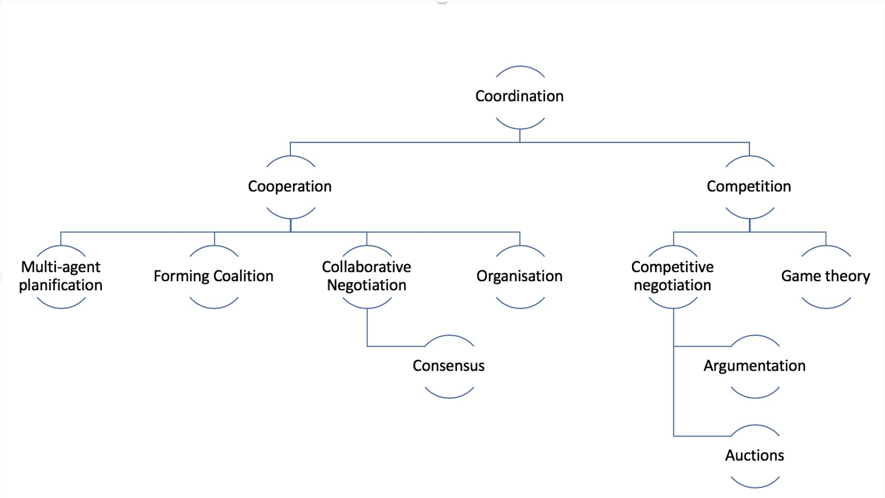
Borrowed from: https://eprints.soton.ac.uk/258850/1/abn.pdf

Conceptual elements of an Argument Based Negotiation agent



Borrowed from: https://eprints.soton.ac.uk/258850/1/abn.pdf

Coordination Mechanisms



Source: borrowed from Aurélie Beynier's course

Conclusion

Pour résumer

Un SMA est généralement caractérisé par :

1. chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent a un point de vue partiel;
2. il n'y a aucun contrôle global du système multi-agent;
3. les donnés sont décentralisées;
4. le calcul est asynchrone.

Pour résumer

- Les SMA possèdent: les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse (avec le parallélisme), et la fiabilité (due à la redondance).
- Ils héritent aussi des bénéfices envisageable de l'IA comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité mais surtout, ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués.
- Les types courants d'interaction incluent la coopération (travailler ensemble à la résolution d'un but commun) ; la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées) ; et la négociation (parvenir à un accord acceptable pour toutes les parties concernées).

Pour résumer

Les questions autour des SMA

1. Comment permettre aux agents de communiquer et d'interagir ? Quoi et quand communiquer?
2. Comment assurer que les agents agissent de manière cohérente i) en prenant leurs décisions ou actions, ii) en gérant les effets non locaux de leurs décisions locales et iii) en évitant les interactions nuisibles ?
3. Comment permettre aux agents individuels de représenter et raisonner sur les actions, plans et connaissances des autres agents afin de se coordonner avec eux ?
4. Comment reconnaître et réconcilier les points de vue disparates et les intentions conflictuelles dans un ensemble d'agents essayant de coordonner leurs actions ?
5. Comment trouver le meilleur compromis entre le traitement local au niveau d'un seul agent et le traitement distribué entre plusieurs agents (traitement distribuée qui induit la communication) ? Plus généralement, comment gérer la répartition des ressources limitées ?
6. . . .

Pour résumer

- Les SMA sont à l'intersection de plusieurs domaines scientifiques : informatique répartie et génie logiciel, intelligence artificielle, vie artificielle.
- Ils s'inspirent également d'études issues d'autres disciplines connexes notamment la sociologie, la psychologie sociale, les sciences cognitives et bien d'autres.
- C'est ainsi qu'on les trouve parfois à la base des : systèmes distribués; interface personnes-machines ; bases de données et bases de connaissances distribuées coopératives; systèmes pour la compréhension du langage naturel, protocoles de communication et réseaux de télécommunications, robotique cognitive et coopération entre robots, applications distribuées comme le web, l'Internet, le contrôle de trafic routier, le contrôle aérien, les réseaux d'énergie, etc;

Using of messaging communication in Mesa

open and explore mesa.zip

Create two communicating agents named Alice and Bob. Create a third agent named Charles whose role is hold a variable v and process messages from Alice and Bob:

- On their turn, Alice and Bob ask Charles for the value of v, using a message;
- If the value is different from their preferred value, they send a message to Charles to change the value of v;
- On its turn, Charles reads its mailbox and processes all messages:
  - Messages that request information about v produce an anwer;
  - Messages that request a change to v are applied.

Implement the Alice-Bob-Charles example.