Multi-agents

Interaction

Interaction

- ***Introduction***
  - Definitions, principle,
  - The interaction components .
- ***Interaction models typology***
  - Direct Interaction,
  - Indirect Interaction
- ***Agent communication language***
  - KQML, FIPA-ACL,
- ***Interaction protocols***
  - Models for the interaction protocols,
  - Instance of protocols.

Introduction

Definition

- **General**
  - **Interaction** is a kind of action that occurs as two or more objects have an effect upon one another. (*wikipedia*)
  - [Morin 77] : Interactions are reciproquable actions modifying the behavior or the nature of the elements, bodies, objects, phenomenas being in presence or in influence.
    - *Les interactions sont des actions réciproques modifiant le comportement ou la nature des éléments, corps, objets, phénomènes en présence ou en influence.*
- **Several points of view**
  - *Models of interaction*: how the agents interact ?
  - *Support of interaction*: how the agents can interact ?
  - *Interaction Engineering*: how interactions are modeled and can be combined?
  - *Interaction modeling*: how interactions influence the agent behavior ?
  - …

[Morin 77] Morin, E. La methode. Tome 1. La nature de la nature. Editions du Seuil, Paris, 1977.

Introduction

Interaction modalities

- ***Without exchange*: the agents cannot explicitly exchange information and their reasoning process is only based on the information they perceive.**
- ***Through a shared space*: the agents perceive the information put by the others.**
- ***Information exchange* : the agents exchange information (simple signals, plan(their tasks and believes), Messages (intention and needs).**

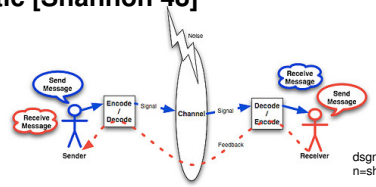
Interaction

- ***Introduction***
  - Definitions, principle,
  - The interaction components .
- ***Interaction models typology***
  - Direct Interaction,
  - Indirect Interaction
- ***Agent communication language***
  - KQML, FIPA-ACL,
- ***Interaction protocols***
  - Models for the interaction protocols,
  - Instance of protocols.

Introduction

Direct interaction

- **Problematic [Shannon 48]**



dsgnfordelight.com/blog.php?subactio n=showcomments&id=1381095141

- **Definition : Communication is the intentional exchange of information brought about by the production and perception of signs drawn from a shared system of conventional signs. [Russel et Norvig 03]**

[Russel et Norvig 03] Russell, S. J. et Norvig, P. Articial Intelligence : a modern approach. Prentice Hall, Upper Saddle River, N.J., second international edition edition, 2003.
[Shannon 48] Shannon, C. E. A mathematical theory of communication. The Bell System Technical Journal, 27(3) :379-423, 1948.

Direct Interaction

Problematics

- **Interpretation problem**
  - How to ensure the correct interpretation of the messages?
  - communication languages, content language,…
- **Conversation problem**
  - How is managed the succession of messages?
  - communication protocols, communication languages,…
- **Connection problem**
  - How to find the right receiver?
  - middle-agent, protocols, platform,…
- **Openness problem**
  - How to maintain the knowledge for interaction?
  - middle-agent, protocols, platform,…
- **…**

Direct interaction

Connection Problem

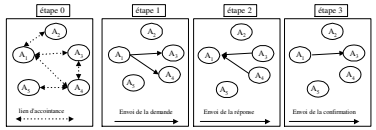
- **Problematic: With which agent I should interact to get a service, a resource, …?**
- **Solutions**
  - Management of social knowledge
    - At agent level: **Acquaintances**,
    - At multi-agent level : **middle-agent**, organization,
    - At platform level: yellow/white pages
  - Protocols
    - **Contract net protocol [Davis and Smith 83]**,
    - Matchmaker, broker
- **Issues**
  - **What is the cost of the solution (number of messages, processing) ?**
  - **Is the solution simple to apply?**
  - **Is the research complete?**

[Davis and Smith 83] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. Artificial Intelligence , 20(1):63–109, January 1983.

Direct interaction

Acquaintances

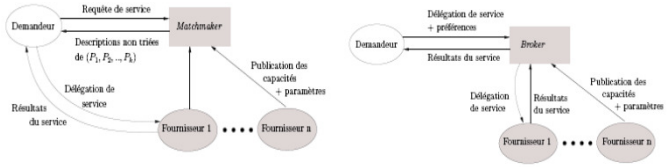
- **Principle : The solution is based on the social knowledge of the agents i. e. their acquaintances.**



- **Advantages**
  - Simplicity,
  - The number of messages is a priori limited.
- **Limits**
  - The dynamic (openness and the state of the agents) management of the social knowledge.
  - The research space is limited to the knowledge of the agent.

Direct Interaction

Middle-Agent [Sycara 00]



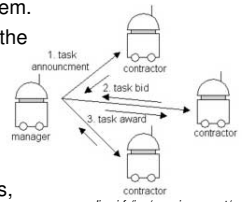
- **Principle : specialized agent to record the social knowledge.**
- **Advantages**
  - A solution to the openness problem
  - The number of messages is limited
  - Other services can be combined (anonymization, selection, ...)
- **Limits**
  - The dynamicity of the information,
  - The centralization of the service

[Sycara 00] Sycara, K. et Wong, H. A taxonomy of middle-agents for the internet. Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000), pages 465-466. IEEE Computer Society, Washington, DC, USA, 2000.

Direct interaction

Contract Net Protocol

- **Principle : the initiator sends/broadcasts a request for a task, resource, … and selections the best bid.**
- **Advantages**
  - Simplicity,
  - A potential solution to the openness problem.
  - The sender and receivers are involved in the interaction process.
  - A distributed solution



perso.limsi.fr/jps/enseignement/examsma/2004/GRAMOLI/html

- **Limits**
  - A potential important number of messages,
  - Several not useful processings,
  - Concurrency management

Indirect Interaction

Principle

- **Definition**

Indirect interaction is interaction via *persistent, observable state changes*; destinations are any agents that will observe these changes. [Keil 2003]

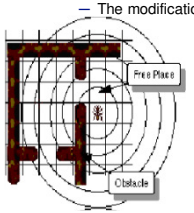
- **Modalities:**
  - **Stigmergy**: modification of the environment,
  - Shared spaces: BlackBoard, **tuple space.**

[Keil 2003] Keil, D., & Goldin, D. (2003, June). Modeling indirect interaction in open computational systems. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on* (pp. 371-376). IEEE.

Indirect Interaction

Stigmergy

- **Principle**
  - The communication between agents is the result of their modification of the environment.
- **Origin**
  - P.P. Grassé 59, La théorie de la Stigmergie : Essai d'interprétation du comportement des termites constructeurs, Insectes Sociaux, 6, 1959, p. 41-80.
  - Example:
    - The use of pheromones,
    - The modification of the spiderweb by social spiders




Indirect Interaction

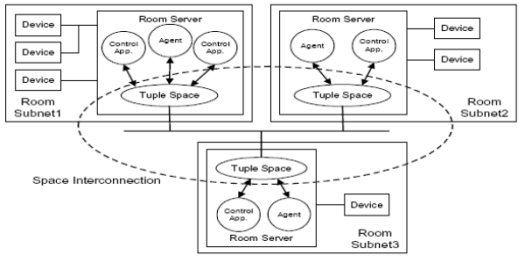
Tuples space

- **Origin : distributed systems.**
- **Principle [Carriero 86] : The Linda model proposes a shared memory called tuples space and a data recovering mechanism based on signature**
- **Implementation:**
  - A tuple is an ordered list of typed data,
  - A template is a tuple where the fields are typed but are not mandatory valued,
  - Three operators:
    - *out(t)* : add the tuple *t*,
    - *in(m)* : retract to read the tuple associated to the template *m*
    - *read(m)* : read the tuple associated to the template *m*

[Carriero 86] Carriero, N., Gelernter, D., et Leichter, J. Distributed data structures in linda. Dans popl'86 :Proceedings of the 13th ACM Sigact-Sigplan symposium on Principles Of Programming Languages, pages 236{242. 1986.

Tuple space

Example



Lee, M. J., Park, J. H., Kang, S. J., and Lee, J. B. 2004. Multi-agent based home network management system with extended real-time tuple space. In Proceedings of the 17th international Conference on innovations in Applied Artificial intelligence (Ottawa, Canada, May 17 - 20, 2004). R. Orchard, C. Yang, and M. Ali, Eds. Lecture Notes In Computer Science. Springer Springer Verlag, 188-198. DOI= http://dx.doi.org/10.1007/b97304

Interaction

- ***Introduction***
  - Definitions, principle,
  - The interaction components .
- ***Interaction models typology***
  - Direct Interaction,
  - Indirect Interaction
- ***Agent communication language***
  - KQML, FIPA-ACL,
- ***Interaction protocols***
  - Models for the interaction protocols,
  - Instance of protocols.

Agent Communication Language

Introduction

- **Initial hypothesis : A common language is an interface between agents.**
  - Syntax: defined how the symbols are structured,
  - Semantic: defined the meaning of the used symbols,
  - The messages are ordered.
- **Sources**
  - Speech act theory [Austin 62, Searle 72, Vanderveken 88]

[Austin 62] J. L. Austin, How to do the things with words. Cambridge, Cambridge University Press, 1962
[SEARLE 72] SEARLE J., *Les actes de langage*, Paris, Hermann, 1972
D. Vanderveken, *Meaning and speech acts*, Cambridge University Press, 1990.

Agent Communication Language

Speech Act

- **Principle :**
  - Verbal actes are called speech acts, they are the building blocks of natural language.
- **Taxonomy (Searle 1969)**
  - *representatives:* such as *informing*, e.g., 'It is raining'
  - *directives:* attempts to get the hearer to do something e.g., 'please make the tea'
  - *commisives:* which commit the speaker to doing something, e.g., 'I promise to… '
  - *expressives:* whereby a speaker expresses a mental state, e.g., 'thank you!'
  - *declarations:* such as declaring war or christening

[SEARLE 69] Searle, J., *Speech Acts, An Essay in the Philosophy of Language*, Cambridge, Cambridge University Press, 1969.

Agent Communication Language

speech Act

- **Pour chaque acte :**
  - "*Locutionary act:* the simple speech act of generating sounds that are linked together by grammatical conventions so as to say something meaningful. Among speakers of English, for example, 'It is raining' performs the locutionary act of saying that it is raining, as 'Grablistrod zetagflx dapu' would not."
  - "*Illocutionary act:* the speech act of doing something else – offering advice or taking a vow, for example – in the process of uttering meaningful language. Thus, for example, in saying 'I will repay you this money next week,' one typically performs the illocutionary act of making a promise."
  - "*Perlocutionary act:* the speech act of having an effect on those who hear a meaningful utterance. By telling a ghost story late at night, for example, one may accomplish the cruel perlocutionary act of frightening a child."

Agent Communication Language

Message

- **Every multiagent language contains at least the following fields:**
  - Sender
  - Receiver
  - Language used
  - Ontology
  - Content
- **There is some debate about whether this (or any!) typology of speech acts is appropriate**
  - In general, a speech act can be seen to have two components:
    - a *performative verb:* (e.g., request, inform, promise, …)
    - *propositional content:* (e.g., "the door is closed")

Agent Communication Language

KQML

- **Knowledge Query and Manipulation language was developed by the ARPA knowledge sharing initiative.**
- **KQML is comprised of two parts:**
  - the knowledge query and manipulation language (KQML)
    - An 'outer' language, that defines various acceptable 'communicative verbs', or *performatives*
  - the knowledge interchange format (KIF)
    - a language for expressing message *content*

KQML

Syntax

Message level
(KQML-performative
:language <text>
:ontology <text>

Communication level
:sender <text>
:receiver <text>

Content level
:content < expression>

```
(tell       :language     prolog
            :ontology     Genealogy
            :in-reply-to  q1
            :sender       Gen-1
            :receiver     Gen-DB
            :content      ''father(John,Alice)'')
```

Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994, November). KQML as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management* (pp. 456-463). ACM.

KQML

performatives

Basic query performatives:
- evaluate, ask-if, ask-in, ask-one, ask-all, ...

Multi-response query performatives:
- stream-in, stream-all, ...

Response performatives:
- reply, sorry, ...

Generic informational performatives:
- tell, achieve, cancel, untell, unachieve, ...

Generator performatives:
- standby, ready, next, rest, discard, generator, ...

Capability-definition performatives:
- advertise, subscribe, monitor, import, export, ...

Networking performatives:
- register, unregister, forward, broadcast, route, ...

KQML

Semantic

- **Bel,** as in bel(A,P) which has the meaning that P is true for A. P is an expression in the native language of A's application (P "exists" in the agent's knowledge base (or virtual knowledge base)).
- **Know,** like the following two operators, refers to the cognitive state of the agents. Know(A,P) expresses a state of knowledge awareness on behalf of A, about P.
- **Want**, as in want(A,P), to mean that agent A desires the event (or state) described by P, to occur.
- **Intend**, as in intend(A,P), to mean that A has every intention of doing P.

Labrou, Y., & Finin, T. (1994, November). A semantics approach for KQML—a general purpose communication language for software agents. In *Proceedings of the third international conference on Information and knowledge management* (pp. 447-455). ACM.

Table 7.2: KQML performatives.

Performative	Meaning
achieve	S wants R to make something true of their environment
advertise	S claims to be suited to processing a performative
ask-about	S wants all relevant sentences in R's VKB
ask-all	S wants all of R's answers to a question C
ask-if	S wants to know whether the answer to C is in R's VKB
ask-one	S wants one of R's answers to question C
break	S wants R to break an established pipe
broadcast	S wants R to send a performative over all connections
broker-all	S wants R to collect all responses to a performative
broker-one	S wants R to get help in responding to a performative
deny	the embedded performative does not apply to S (any more)
delete-all	S wants R to remove all sentences matching C from its VKB
delete-one	S wants R to remove one sentence matching C from its VKB
discard	S will not want R's remaining responses to a query
eos	end of a stream response to an earlier query
error	S considers R's earlier message to be malformed
evaluate	S wants R to evaluate (simplify) C
forward	S wants R to forward a message to another agent
generator	same as a standby of a stream-all
insert	S asks R to add content to its VKB
monitor	S wants updates to R's response to a stream-all
next	S wants R's next response to a previously streamed performative
pipe	S wants R to route all further performatives to another agent
ready	S is ready to respond to R's previously mentioned performative
recommend-all	S wants all names of agents who can respond to C
recommend-one	S wants the name of an agent who can respond to a C
recruit-all	S wants R to get all suitable agents to respond to C
recruit-one	S wants R to get one suitable agent to respond to C
register	S can deliver performatives to some named agent
reply	communicates an expected reply
rest	S wants R's remaining responses to a previously named performative
sorry*	S cannot provide a more informative reply
standby	S wants R to be ready to respond to a performative
stream-about	multiple response version of ask-about
stream-all	multiple response version of ask-all
subscribe	S wants updates to R's response to a performative
tell	S claims to R that C is in S's VKB
transport-address	S associates symbolic name with transport address
unregister	the deny of a register
untell	S claims to R that C is *not* in S's VKB

KQML

Semantic

- **For each performative**
  1. A **natural language description** of the performative's intuitive meaning.
  2. An **expression in their logic** that describes the illocutionary act. For all practical purposes, this is a formal representation of the natural language description.
  3. **Preconditions** that indicates the necessary state for an agent in order to send a performative and for the receiver to accept it and process it.
  4. **Postconditions** that describe the state of agents after the utterance of a performative (for the sender) and after the receipt (but before a counter utterance) of a message (by the receiver)
  5. **Completion conditions** for the sender that indicate the final state of the sender, after possibly a conversation has taken place and the intention suggested by the performative that started the conversation, has been fulfilled.
  6. **Any natural language comments** that we might find suitable to enhance the understanding of the performative.

KQML

Example

Tell(A,B,X)

1. **Natural expression**
   - A states to B that A believes the content X is true.
2. **Expression in their logic**
   - bel(A,X)
3. **Precondition :**
   - Pre(A): bel(A,X) , know(A,want(B,know(B,Y)))
     - A does not lie and A know that B is interested in knowing Y.
   - Pre(B): intend(B,know(B,Y))
4. **PostCondition :**
   - Post(A): know(A,know(B,bel(A,X))) (optional)
   - Post(B): know(B,bel(A,X))
5. **Completion**
   - Completion(A): know(B,bel(A,X))
     - The completion condition holds, unless a sorry or error suggests B's inability to acknowledge properly the tell.

KQML

Discussion

- **Advantages**
  - First communication "standard"
  - Numerous applications supported KQML
  - Extensible language:
    - New performatives can be created
    - New parameters can be added
    - Takes into account ontologies
- **Limits**
  - Several implementations were developed that could not interoperate
  - Semantic has never been rigorously defined and it is never sure that agents use KQML correctly
  - The entire class of performative *commissive* (an agent make a commitment to another) is missing. These performatives are essential for coordination.
  - There are too many performatives to be efficient.
  - Some KQML performatives are not considered as real performatives because they are used for mediation (e. g. recruit) or networking actions (e. g. broadcast or forward)
  - Does not take into account conversation

FIPA-ACL

Introduction

- **FIPA = Foundation for Intelligent Physical Agents**
- **FIPA-ACL benefits of the research about KQML**
- **FIPA-ACL is superficially similar to KQML**

	Parameter	Category of Parameters
*	performative	Type of communicative acts
	sender	Participant in communication
*	receiver	Participant in communication
	reply-to	Participant in communication
	content	Content of message
	language	Description of Content
	encoding	Description of Content
	ontology	Description of Content
	protocol	Control of conversation
	conversation-id	Control of conversation
	reply-with	Control of conversation
	in-reply-to	Control of conversation
	reply-by	Control of conversation

FIPA-ACL

Performative

Table 7.3: Performatives provided by the FIPA communication language.

Performative	Passing information	Requesting information	Negotiation	Performing actions	Error handling
accept-proposal			×	×	
agree		×		×	
cancel			×		
cfp					
confirm	×				
disconfirm	×				×
failure					
inform	×				
inform-if	×				
inform-ref	×				×
not-understood				×	
propagate			×		
propose				×	
proxy					
query-if		×			
query-ref		×		×	
refuse			×		
reject-proposal				×	
request				×	
request-when				×	
request-whenever				×	
subscribe		×			

FIPA-ACL

Performative

- accept-proposal: the action of accepting a previously submitted proposal to perform an action
- agree: the action of agreeing to perform some action, possibly in the future
- cancel: the action of cancelling some previously requested action which has temporal extent
- cfp: the action of calling for proposals to perform a given action
- confirm: the sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition
- disconfirm: the sender informs the receiver that a given poposition is false, where the receiver is known to believe, or believe it likely that, the proposition is true.
- failure: the action of telling another agent that an action was attempted but the attempt failed.
- propagate: the sender intends that the receiver treats the embedded message as sent directly to it, and wants the receiver to identify the agents denoted by the given descriptor and send the received propagate message to them
- propose: the action of submitting a proposal to perform a certain action, given certain preconditions
- proxy: the sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them
- query-if: the action of asking another agent whether or not a given proposition is true.
- query-ref:the action of asking another agent for the object referred to by a referential expression
- refuse: the action of refusing to perform a given action and explaining the reason for the refusal.
- reject-proposal: the action of rejecting a proposal to perform some action during a negotiation.
- request: the sender requests the receiver to perform some action.
- request-when: the sender wants the receiver to perform some action when some given proposition becomes true.
- request-whenever: the sender wants the receiver to perform some action as soon as some proposition is true and thereafter each time the proposition becomes true again.
- subscribe: the act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes.

FIPA-ACL

Semantic

- **The semantics of the FIPA ACL maps each ACL message to a formula of a formal language called SL**
  - Represents beliefs ($B_i(p)$), desire ($I_i(p)$) and uncertain beliefs of agents ($Bif_i(p)$, $Uif_i(p)$) as well as actions that agents perform.
- **The operators *Feasible*, *Done* and *Agent* are introduced to enable reasoning about actions, as follows:**
  - Feasible (*a*, *p*) means that *a* can take place and if it does *p* will be true just after that,
  - Done (*a*, *p*) means that *a* has just taken place and *p* was true just before that,
  - Agent (*i*, *a*) means that *i* denotes the only agent that ever performs (in the past, present or future) the actions which appear in action expression *a*,

http://www.fipa.org/specs/fipa00037/SC00037J.html

FIPA-ACL

Semantic

- **"Inform" and "Request" are the two basic performatives in FIPA. All others are *macro* definitions, defined in terms of these.**
- **The meaning of inform and request is defined in two parts:**
  - Feasibility Preconditions
    - what must be true in order for the speech act to succeed
    - Conformance requires the sender respects the feasibility preconditions.
  - Rational Effect
    what the sender of the message hopes to bring about
    - Conformance does not require the recipient of a message to respect the rational effect.

http://www.fipa.org/specs/fipa00037/SC00037J.html

FIPA-ACL

Example

- **<i, inform(k,p)>**
  - FP : $B_i p \wedge \neg B_i(Bif_k p \vee Uif_k p)$
    - *i* believes *p* ($B_i p$) , and it is not the case ($\neg B_i()$) that it believes of *k* either that *k* believes whether *p* is true or false ($Bif_K p$), or that j is uncertain of the truth or falsity of p ($Uif_k p$).
  - RE : $B_k p$
    - If the agent is successful in performing the *inform* then the receiver (agent k) will believe *p*
- **<i, request(k, p)>**
  - FP : B_iAgent (p,k) $\wedge \neg B_i I_k$Done(p)
    - Agent(p,k): the agent k is the agent that can perform p
    - Done(p): the action p has been done
    - Agent *i* believes that *k* is the agent that performs *p* and agent *i* believes that agent *k* does not currently intend that *p* is done.
  - RE: Done(p)

FIPA-ACL

Conversation

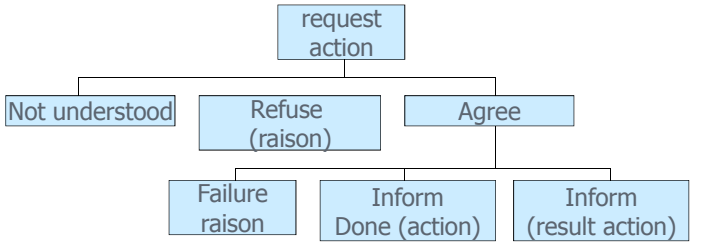
- **Each speech act is associated to a protocol**
  - The sender of a message knows which kind of answer he will receive
  - The receiver knows how he must respond to each received message

```
(inform
    :sender A
    :receiver B
    :content (price (bid goood02) 150)
    :in-reply-to round-4
    :reply-with bid04
    :envelope 1000
    :language s1
    :ontology hpl-auction
    :reply-by 10
    :protocol offer
    :conversation-id conv02
)
```


FIPA – ACL

Protocol Example

FIPA Request protocol



Interaction

- ***Introduction***
  - Definitions, principle,
  - The interaction components .
- ***Interaction models typology***
  - Direct Interaction,
  - Indirect Interaction
- ***Agent communication language***
  - KQML, FIPA-ACL,
- ***Communication protocols***
  - Models for the communication protocols,
  - Instance of protocols.

Protocol

Introduction

- **Definition**
  - An agent communication protocol describes:
    - a *communication pattern*, with the allowed sequence of messages between agents having different roles,
    - some (semantics) restrictions on the content of the messages,
    - the semantics according to the semantics of the speech acts, i.e. the use of speech acts within the pattern has to be consistent with their semantics.
- **Issues**
  - Formalism to design protocols,
  - Definition of protocols

Formalisms for protocols

Finite state machine

- **Example Cool [Barbuceanu 95]**
  - The states of the FSM represent the *states* a conversation can be in. There is a distinguished *initial* state any conversation starts in, and several *terminating* states that when reached signal the termination of the conversation.
  - The messages exchanged are represented as *performatives* (speech acts) of the agent communication language.
  - A set of *conversation rules* specify how an agent in a given state receives a messages of specified type, does local actions (e.g. updating local data), sends out messages, and switches to another state.
  - A set of *error recovery rules* specify how incompatibilities among the state of a conversation and the incoming messages are handled.
  - A set of *continuation rules* specify how agents accept requests for new conversations or select a conversation to continue from among the existing ones.
  - *Conversation classes* specify the states, conversation rules and error rules that are specific to a type of conversation. An agent has several conversation classes it can use when communicating with other agents.
  - *Actual conversations* instantiate conversation classes and are created whenever agents engage in communication

Barbuceanu, M., & Fox, M. S. (1995, June). COOL: A Language for Describing Coordination in Multi Agent Systems. In *ICMAS* (pp. 17-24).

Cool

Example

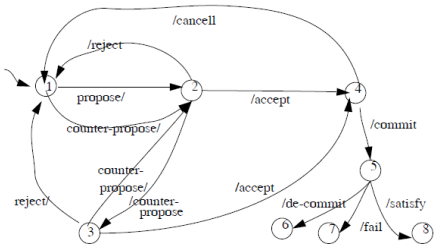


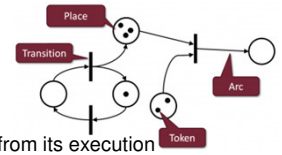
FIGURE 1. State transitions for negotiation.

Barbuceanu, M., & Fox, M. S. (1995, June). COOL: A Language for Describing Coordination in Multi Agent Systems. In *ICMAS* (pp. 17-24).

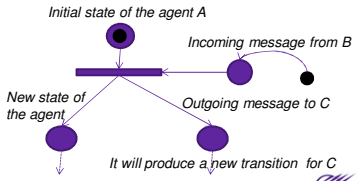
Formalisms for protocols

Petri Net

- **A petri net is a graph which**
  - contains 2 types of nodes
    - Circles (Places) :
    - Bars (Transitions)
  - has dynamic properties that result from its execution
    - Markers (Tokens)
    - Tokens are moved by the firing of transitions of the net.

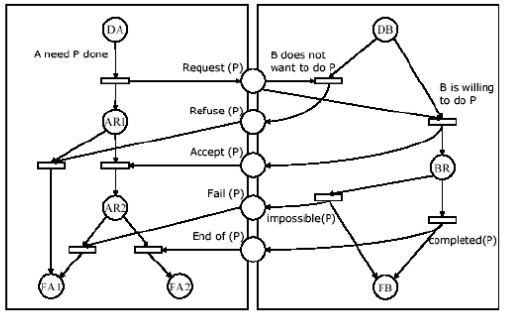


- **A Multiagent point of view**
  - Places
    - Internal state of the agents
    - Specific messages
  - Transitions
    - Reception of messages,
    - Agent actions.



Petri Net

Example



www.emse.fr/~boissier/enseignement/maop12/courses/iop-4pp.pdf

Formalisms for protocols

AUML

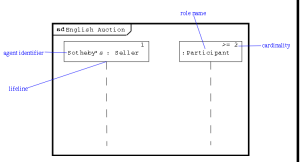
- **AUML (agent-based unified modeling language) is the result of a working group of the FIPA.**
- **The objectives are:**
  - Gain an insight into how agent-oriented software engineering can benefit from UML and other modeling languages.
  - Focus on problems and notations that are deemed necessary to support modeling of autonomous agents systems
  - Adopt notations that graphically express various aspects of agent-base modeling by extending UML and/or by using other notations.
  - Address standards for AUML class and sequence diagrams

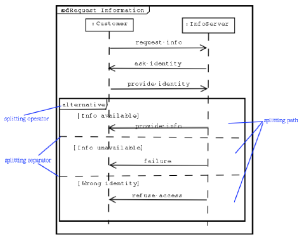
www.fipa.org/docs/wps/f-wp-00022/f-wp-00022.html

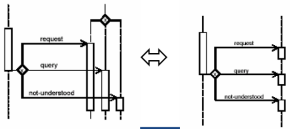
AUML

Representation

- **Two dimensions**
  - Vertical: temporal representation of the protocol (lifeline)
    - The sequence of messages is ordered.
  - Horizontal: organizational representation of the protocol
    - The role of the participants to the protocol.
- **Alternative (or operator)**







www.cin.ufpe.br/~in1096/2006-1/AUML/AOSE-Bauer[B-I].pdf

www.auml.org/auml/projects/main.shtml

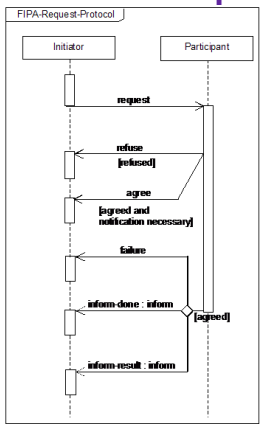
Protocol
FIPA – Interaction Protocol

- **Notation : AUML**
- **Existing protocols**

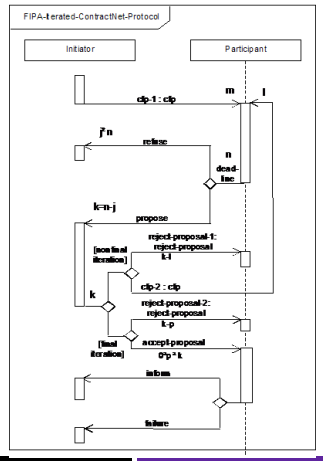
Identifier	Title
SC00026	FIPA Request Interaction Protocol Specification
SC00027	FIPA Query Interaction Protocol Specification
SC00028	FIPA Request When Interaction Protocol Specification
SC00029	FIPA Contract Net Interaction Protocol Specification
SC00030	FIPA Iterated Contract Net Interaction Protocol Specification
XC00031	FIPA English Auction Interaction Protocol Specification
XC00032	FIPA Dutch Auction Interaction Protocol Specification
SC00033	FIPA Brokering Interaction Protocol Specification
SC00034	FIPA Recruiting Interaction Protocol Specification
SC00035	FIPA Subscribe Interaction Protocol Specification
SC00036	FIPA Propose Interaction Protocol Specification

www.fipa.org/repository/ips.php3

Protocol
Fipa - request



Protocol
CNP



Protocol
brokering fipa request