**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 reciprocable 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?
  - …


---

**Interaction**

- **Interaction models typology**
  - Direct Interaction,
  - Indirect Interaction

- **Agent communication language**
  - KQML, FIPA-ACL,

- **Interaction protocols**
  - Models for the interaction protocols,
  - Instance of protocols.

---

**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.

---

**Direct Interaction**

**Problematic [Shannon 48]**

**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.


---

**Direct Interaction**

**Problematic**

- 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?

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

**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

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

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

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.
**Indirect Interaction**

**Stigmergy**

- **Principle**
  - The communication between agents is the result of their modification of the environment.

- **Origin**
  - Example:
    - The use of pheromones,
    - The modification of the spider web by social spiders

**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

**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]

---

**Agent Communication Language**

**Speech Act**

- **Principle**: Verbal acts 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

---

**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")
KQML 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

**Syntax**

- **Message level**
  - `:language <text>`
  - `:ontology <text>`
- **Communication level**
  - `:sender <text>`
  - `:receiver <text>`
- **Content level**
  - `:content <expression>`

**Performatives**

- **Basic query performatives**
  - `evaluate`, `ask if`, `ask in`, `ask all`...
- **Multi-response query performatives**
  - `stream in`, `stream all`...
- **Response performatives**
  - `reply`, `sorry`...
- **Generic informational performatives**
  - `tell`, `achieve`, `cancel`, `until achieve`...
- **Generator performatives**
  - `assemble`, `ready`, `reset`, `discard`, `generator`...
- **Capability-definition performatives**
  - `advise`, `subscribe`, `monitor`, `import`, `export`...
- **Networking performatives**
  - `register`, `unregister`, `forward`, `broadcast`, `route`...

**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.
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.

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,bel(B,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.

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
Introduction

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

<table>
<thead>
<tr>
<th>Performative</th>
<th>Category of Transmitters</th>
</tr>
</thead>
<tbody>
<tr>
<td>performative</td>
<td>Tips of communicative acts</td>
</tr>
<tr>
<td>execute</td>
<td>Participant in communication</td>
</tr>
<tr>
<td>reply-to</td>
<td>Participant in communication</td>
</tr>
<tr>
<td>context</td>
<td>Content of message</td>
</tr>
<tr>
<td>language</td>
<td>Description of Content</td>
</tr>
<tr>
<td>encoding</td>
<td>Description of Content</td>
</tr>
<tr>
<td>perform</td>
<td>Control of conversation</td>
</tr>
<tr>
<td>conversationID</td>
<td>Control of conversation</td>
</tr>
<tr>
<td>reply-with</td>
<td>Control of conversation</td>
</tr>
<tr>
<td>reply-by</td>
<td>Control of conversation</td>
</tr>
</tbody>
</table>

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 canceling some previously requested action which has temporal scope.
- execute: the action of doing the requested action.
- confirm: the sender obtains the receiver to a given proposition is true, where the receiver is known to be uncertain about the proposition.
- disconfirm: the sender obtains the receiver to a given proposition is false, where the receiver is known to be uncertain about the proposition.
- deny: the action of telling another agent that an action was attempted but the attempt failed.

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/fipa00037/00037J.html
"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

Example

\(<i, \text{inform}(k, p)>)\)
- \(\text{FP} : B_i p \land \neg (B_i (B_k p \lor U_k p))\)
- \(i\) believes \(p\), and it is not the case (\(\neg B_i()\)) that it believes of \(k\) either that \(k\) believes whether \(p\) is true or false (\(B_k p\)), or that \(j\) is uncertain of the truth or falsity of \(p\) (\(U_k p\)).
- \(\text{RE} : B_k p\)
  - If the agent is successful in performing the inform then the receiver (agent \(k\)) will believe \(p\)

\(<i, \text{request}(k, p)>)\)
- \(\text{FP} : B_j (\text{Agent}(p,k) \land \neg B_k \text{Done}(p))\)
  - \(\text{Agent}(p,k)\): the agent \(k\) is the agent that can perform \(p\)
  - \(\text{Done}(p)\): the action \(p\) has been done
  - \(i\) believes that \(k\) is the agent that performs \(p\) and \(i\) believes that agent \(k\) does not currently intend that \(p\) is done.
- \(\text{RE} : \text{Done}(p)\)

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

```
(iInform
  :sender A
  :receiver B
  :content price (bid good02) 150
  :in-reply-to round-4
  :reply-with bid-04
  :envelope 1000
  :language s1
  :ontology rdf-auction
  :reply-by 10
  :protocol offer
  :conversation id conv02
)
```

FIPA – ACL

Protocol Example

FiPA Request protocol

- request
- action

- Not understood
- Refuse (raison)
- Agree

- Failure
- raison
- Inform
- Done (action)
- Inform (result action)
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 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.

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.

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

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 in the protocol

Alternative (or operator)
Protocol
FIPA – Interaction Protocol

- Notation: AUML
- Existing protocols

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC00025</td>
<td>FIPA Request Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00027</td>
<td>FIPA Query Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00028</td>
<td>FIPA Request When Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00030</td>
<td>FIPA Iterated Contract Net Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00031</td>
<td>FIPA English Auction Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00032</td>
<td>FIPA Dutch Auction Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00033</td>
<td>FIPA Brokering Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00034</td>
<td>FIPA Recruiting Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00035</td>
<td>FIPA Subscribe Interaction Protocol Specification</td>
</tr>
<tr>
<td>SC00036</td>
<td>FIPA Propose Interaction Protocol Specification</td>
</tr>
</tbody>
</table>

www.fipa.org/repository/ips.php3

Protocol
Fipa - request

Protocol
CNP

Protocol
brokering fipa request