2. Knowledge Representation and Communication

Part 2: Agent Communication

Javier Vázquez-Salceda
SMA-UPC

Why agent communication?

- In order to solve distributed problems, agents need to coordinate (cooperate, compete) with others.
- For this, agents need to communicate.
- Goals for Agent Communication:
  - Agents able to request (to other agents) actions or services they cannot perform by themselves.
  - Agents able to ask for information (to other agents).
  - Agents able to share their beliefs with other agents.
  - Agents able to coordinate with other agents. To solve complex tasks.
Levels in Agent Communication

- Four levels in communication:
  
  **Message Semantics**
  - What does each message means?
  - 3 components
    - **Message type**: gives intensionality
    - **Message content**: contains the information
    - **Ontology** (the message refers to)
  
  **Message Sintaxis**
  - How each message is expressed?
  - 2 components
    - Message structure: **Agent Communication Language**
    - Content codification: **Content Language**
  
  **Interaction protocol**
  - How are conversations/dialogues structured?
    - **Agent Protocols**
  
  **Transport protocol**
  - How messages are actually sent and received by agents?

- message_semantics
- communication
- jvazquez@lsi.upc.edu
- knowledge_representation_and_communication
- levels_in_agent_communication
- multiagent_systems
- agent_communication_language
- content_language
- agent_protocols
- transport_protocol
- speech_act_theory
- https://kemlg.upc.edu
Message Semantics: Speech Acts

- The analysis of the different types of messages that 2 individuals can exchange is within the area of linguistics, and more concretely, *speech act theory*.
- Speech act theories are *pragmatic* theories of language, i.e., theories of language use
  - they attempt to account for how language is used by people every day to achieve their goals and intentions
- *In “How to Do Things with Words” (1962), Austin noticed that some utterances are rather like ‘physical actions’ that appear to change the state of the world*
- Paradigm examples would be:
  - declaring war
  - christening
  - ‘I now pronounce you man and wife’
- But more generally, *everything* we utter is uttered with the intention of satisfying some goal or intention

Speech Acts

Aspects

- *Locutionary act* or *location*: what it is said or written (the sentence, the sounds).
  - E.g. ‘It is raining’ performs the locutionary act of saying that it is raining.
- *Illocutionary act* or *illocution*: what it is not said or written explicitly, but it is meant.
  - E.g. ‘I will repay you this money next week’ typically performs the illocutionary act of making a promise.
- *Perlocutionary act* or *perlocution*: the effect provoked on those who hear a meaningful utterance.
  - E.g. 1: ‘Shut up!’ usually has an effect on stopping another individual’s utterances
  - E.g. 2: telling a ghost story late at night may accomplish the cruel perlocutionary act of frightening a child.
Speech Acts
Types

- Searle (1969) identified various different types of speech act:
  - **representatives:**
    - such as informing, e.g., ‘It is raining’
  - **directives:**
    - attempts to get the hearer to do something
      e.g., ‘please make the tea’
  - **commissives:**
    - 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

Speech Acts
Components

- 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”)  

- E.g.:
  - performative = request
    content = “the door is closed”
    speech act = “please close the door”
  - performative = inform
    content = “the door is closed”
    speech act = “the door is closed!”
  - performative = inquire
    content = “the door is closed”
    speech act = “is the door closed?”
How does one define the semantics of speech acts? When can one say someone has uttered, e.g., a request or an inform?

Cohen & Perrault (1979) defined semantics of speech acts using the precondition-delete-add list formalism of planning research.

Note that a speaker cannot (generally) force a hearer to accept some desired mental state.

In other words, there is a separation between the illocutionary act and the perlocutionary act.

E.g., semantics for request:

\[ \text{request}(s, h, \phi) \]

**pre:**
- \( s \) believes \( h \) can do \( \phi \)
  - (you don’t ask someone to do something unless you think they can do it)
- \( s \) believes \( h \) believe \( h \) can do \( \phi \)
  - (you don’t ask someone unless they believe they can do it)
- \( s \) believes \( s \) wants \( \phi \)
  - (you don’t ask someone unless you want it!)

**post:**
- \( h \) believes \( s \) believes \( s \) wants \( \phi \)
  - (the effect is to make them aware of your desire)
Message Syntax

- Agent Communication Language

Speech Acts in Agent Communication Langs.

- Agent communication is based in Speech Act Theory
- Agents use a set of pre-defined performatives in order to communicate their intentions
- The performative semantics allow the agent receiving a message to interpret its content in a proper way
- There are two pre-defined performative sets used in Multiagent Systems:
  - **KQML** Knowledge Query and Manipulation Language
  - **FIPA-ACL** Agent Communication Language
KQML

- The first widely-spread ACL was KQML, developed by the ARPA knowledge sharing initiative
- KQML is comprised of two parts:
  - the knowledge query and manipulation language (KQML)
  - the content language (usually KIF)
- KQML is an ‘outer’ language, that defines a quite large set of acceptable ‘communicative verbs’, or **performatives** for:
  - Basic requests (evaluate, ask-one, perform ...)
  - Multiagent requests (stream-in, ...)
  - Responses (reply, sorry, ...)
  - Information (tell, achieve, cancel, ...)
  - Coordination (stand-by, ready, next, ...)
  - Definition of capabilities (advertise, subscribe, ...)
  - Networking (register, forward, broadcast, ...)

KQML Example

```( ask-one
  :sender joan
  :receiver stock-server
  :reply-with IPOD-stock
  :content (PRICE IPOD ?price)
  :language LISP
  :ontology NYSE-TICKS )```
KQML and KIF

- KIF is a language for expressing message content
- E.g.,
  - “The temperature of m1 is 83 Celsius”:
    \[(= \text{temperature} \text{m1} \text{scalar} 83 \text{Celsius})\]
  - “An object is a bachelor if the object is a man and is not married”:
    \[(\text{defrelation} \text{bachelor} (?x) := \text{and} \text{man} ?x \text{not} \text{married} ?x)\]
  - “Any individual with the property of being a person also has the property of being a mammal”:
    \[(\text{defrelation} \text{person} (?x) := \rightarrow \text{mammal} ?x)\]

Example

( tell
  :sender stock-server
  :receiver joan
  :content (= (price IPOD) (scalar 199 Euro))
  :language KIF
  :ontology NYSE-TICKS )

- In literature a short version of KQML/KIF messages is used to specify dialogues:
  - A to B: (ask-if (> (size chip1) (size chip2)))
  - B to A: (reply true)
  - B to A: (inform (= (size chip1) 20))
  - B to A: (inform (= (size chip2) 18))
  - A to B: (perform (print "Hello!" t))
  - B to A: (reply done)
FIPA-ACL

- More recently, the Foundation for Intelligent Physical Agents (FIPA) started work on a program of agent standards — the centrepiece is an ACL.

- Basic structure is quite similar to KQML:
  - Type of communicative act: performative
    - 22 performatives in FIPA (reduction from KQML)
  - Communication actors
    - e.g., sender, receiver.
  - Content
    - the actual content of the message
  - Content description
    - e.g., language, encoding, ontology
  - Conversation control
    - e.g., protocol, conversation-id, reply-with, in-reply-to, reply-by

Example:

```
(inform
  :sender agent1
  :receiver agent5
  :content (price good200 150)
  :language sl
  :ontology hpl-auction
)
```
### FIPA-ACL performatives

<table>
<thead>
<tr>
<th>performatives</th>
<th>passing info</th>
<th>requesting info</th>
<th>negotiation</th>
<th>performing actions</th>
<th>error handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept-proposal</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>agree</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cancel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cfrp</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>confirm</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>disconfirm</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>failure</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inform-if</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inform-ref</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not-understood</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>propose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>query-if</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>query-ref</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>refuse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reject-proposal</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>request-when</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>request-whenever</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>subscribe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### FIPA-ACL performatives for requests

- **request, request-when, request-whenever**: request for an action to be performed unconditionally/when a given condition holds/each time the condition holds
- **propose**: to propose an action to be performed when some given conditions hold
- **call-for-proposal**: request for proposals from other agents to perform actions under certain pre-conditions
- **inform-if, inform-ref, query-if, query-ref**: ask the receiver if he believes that a given condition is true or that for a referred element a given condition holds
- **propagate, proxy**: request another agent to forward a given message, either reading it and propagating it or propagating without reading
- **subscribe**: request to an agent to inform whenever a given expression/object changes its value
FIPA-ACL
performatives for responses

- **inform**: Informs that a given expression is true
- **accept-proposal, reject-proposal**: A proposal (for an action performance) is accepted or rejected
- **confirm, disconfirm**: A fact's truth value is communicated to an agent which has some uncertainty about it
- **agree**: An agreement about performing an action
- **refuse**: A refusal to perform an action (+ reason)
- **cancel**: Cancellation of an agreed action
- **failure**: Action could not be preformed properly
- **not-understood**: Last message has not been understood

FIPA-ACL
Content Language

- Almost any content language can be used with FIPA-ACL. Most used are KIF (ANSI-KIF, ISO-KIF), RDF, DAML, OWL and FIPA-SL
- Others can be used such as PROLOG, SQL, ...
- **FIPA-SL** (Semantic Language)
  - Allows representation of asserts in modal
  - It is designed for agents with BDI architecture (Beliefs, Desires, Intentions)
  - Defines 3 types of content:
    - **Statements**: expressions which can be associated with a truth value
    - **Actions**: expressions defining an action that can be performed
    - **Reference expressions**: quantified formulae referring to domain objects which comply with that formulae
FIPA-SL

Elements

- Expressions in FIPA-SL are in prefix notation (such as in KIF)
- It includes connectives from First Order Logic
  - not, and, or, implies, <=>, forall, exist
- BDI Operators
  - (B <agent> <exp>) Agent believes the expression
  - (U <agent> <exp>) Agent has some uncertainty about the expression
  - (I t) Agent has as an intention the one in the expression
  - (PG <agent> <exp>) Agent has as an objective the one in the expression

Temporal Logic operators

- (feasible <action> <exp>): Action can be performed when expression holds
- (done <action> <exp>): Action was performed before the expression held.

Relational and list operators

- (=, >, <, member, contains)

Reference expressions (evaluated through a Knowledge Base)

- (iota <terms> <exp>): refers to the unique object which, instantiating the terms, makes the expressions true
- (any <terms> <exp>): refers to a/some objects which, instantiating the terms, make the expressions true
- (all <terms> <exp>): refers to all objects which, instantiating the terms, make the expressions true
FIPA-SL

Elements

- Functional Terms (predicates): expressions which refer to an object through its functional relation with other objects (e.g., $3 = (+ 2 1)$). There are two alternative expressions:
  - $(\langle \text{predicate} > \ <\text{value}1> \ ... \ <\text{value}_n>)$, e.g., $(\text{person} \ "\text{Juan}" \ 23)$
  - $(\langle \text{predicate} <\text{prop}1> \ <\text{value}1> \ ... \ <\text{prop}_n> \ <\text{value}_n>)$, e.g., $(\text{person} :\text{name} \ "\text{Juan}" :\text{age} \ 23)$

- FIPASL has some pre-defined functional terms (arithmetic operators, set operators, list operators…)

- Predicates over actions and results
  - $(\text{action} \ <\text{agent}> \ <\text{exp}>)$: we request the agent to perform the action expressed in the expression
  - $(\text{result} \ <\text{action}> \ <\text{exp}>)$: informs about the result of a given action

FIPA-SL

3 subsets

- FIPA-SL defines 3 subsets of the language with different expressiveness, for computational reasons
  - **FIPA-SL0**: Allows predicates action, result, done, simple propositions, sets and sequences
  - **FIPA-SL1**: Adds boolean connectives in expressions
  - **FIPA-SL2**: Adds referential expressions and the modal/temporal operators, but with some restrictions to ensure that the demonstrations are decidable
What are (agent) communication protocols?

- Performatives cannot work alone, but they appear as part of a protocol specification.
- A protocol is a conversation between agents which follows some rules defining which performatives to use and when in order to achieve a given goal.
- Each protocol defines the sequencing of messages in a given dialogue as a finite-state diagram.
- Advantage: agents can easily keep the current state of a dialogue and know which utterances follow in order to comply with the protocol.
- Each protocol is designed for a specific type of dialogue → One should carefully choose which protocol to use for each situation.
Protocols defined by FIPA

- They have two sides: **initiator** and **responder**.

- FIPA protocols: *Request, Query, Contract Net, Iterated Contract Net, Brokering, Recruiting, Subscribe, Propose*

- The most used are:
  - **Request**: dialogue to ask an agent for an action to be performed. The responder agent gives back the result, if possible
  - **Request-When**: dialogue to ask an agent for an action to be performed whenever some conditions hold
  - **Query**: dialogue to ask an agent if a given expression is true. The responder agent answers, if possible
  - **Propose**: dialogue to propose another agent to perform a given action under given conditions. The responder agent accepts or rejects the proposal
  - **Contract Net**: dialogue to request a group of agents to send back proposals for actions to solve a given task. The initiator agent selects the best proposals

FIPA protocols

Request-Response Protocols

- E.g. FIPA specification for **FIPA-Query** and **FIPA-Request**
FIPA protocols
FIPA-Request

FIPA protocols
FIPA-Request-When
FIPA protocols
FIPA-Query

- E.g. FIPA specification for *Contract Net*

- initiator

  - propose (preconditions)

  - refuse (reason)

  - not-understood

  - propose (preconditions)

  - inform (done (action))

- responder

  - accept -proposal (proposal)

  - inform (done (action))

  - failure (reason)

  - cancel (reason)

  - reject -proposal (reason)

  - done (reason)

  - inform (reason)
FIPA protocols

FIPA-Contract-Net (II)
References


These slides are based mainly in material from [2] and from J. Bejar, with some additions from material by U. Cortés and A. Moreno