Based on:
Michael Wooldridge. *Reasoning about Rational Agents.* MIT Press, Cambridge, Mass. 2000. Chapter 7.

Agent Communications

BDI Communication

CPSC 662/CPSC 568
Rob Kremer
Department of Computer Science
University of Calgary

Motivation

- Agents aren't objects – therefore they must *communicate in order to influence* other agents (as opposed to invoking methods on other agents)
- We treat such communications just like other actions, and call it "speech act theory"[1,2]

[1]John Austin. *How to Do Things With Words.* Oxford University Press: Oxford, England, 1962
[2]John Searle, *Speech acts: An Essay in the Philosophy of Language.* Cambridge University Press: Cambridge, England, 1969.

Example

- The state I might want to bring about is that "you believe it's raining in London":

B_{you} Weather(london, raining)

- but just because I send you "Weather(london, raining)" in a message doesn't necessarily mean that you will come to believe it (I may be a compulsive liar, or very bad at weather forecasts)

Speech Acts

- Performative verbs
  - Eg: *request, inform, promise*

- Aspects:
  - Locutionary act: making an utterance
  - Illocutionary act: action performed in saying something
  - Perlocution: the effect of the act
- Felicitory conditions:
  - There exist an accepted procedure, and circumstances and persons must be specified
  - Must be executed correctly and completely
  - Must be sincere, and must have uptake

Example

Request

- Normal I/O conditions:
  - Hearer not deaf
  - Not uttered in a play or film

- Preparatory conditions:
  - Speaker must correctly choose the act
  - Hearer must be able
  - Speaker must believe the hearer is able
  - Hearer wouldn't have done the act anyway
- Sincerity conditions:
  - Not sarcasm, etc.

Types of Speech Acts

- **Representatives**: commits the speaker to the truth of a proposition
  - Eg: Informing
- **Directives**: attempt to get the hearer to do something
  - Eg: Requesting
- **Commissives**: commit to a course of action
  - Eg: Promising
- **Expressives**: express some psychological state
  - Eg: Thanking
- **Declarations**: effects some change in an institutional state of affairs
  - Eg: Declaring war

Rational Action

- *Speech act theory* can be considered as a specialization of a more general *theory of rational action*[1]
- Eg: "A request is an attempt on the part of *spkr*, by doing ε, to bring about a state where, ideally, (i) *addr* intends α (relative to the *spkr* still having that goal, and addr still being helpfully inclined to *spkr*), and (ii) *addr* actually eventually does α, or at least brings about a state where *addr* believes it is mutually believed that it wants the ideal situation" [1,p.241]

[1]P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, and M. E. Polleck (eds), *Intentions in Communications*, pp 221-56. The MIT Press: Cambridge, MA, 1990.

FIPA Messages

performative	the type of the communicative act of the message
sender	the sender of the message
receiver	the intended recipients of the message
reply-to	subsequent messages in this conversation thread are to be directed to the agent named
content	the content of the message; equivalently denotes the object of the action
language	the language in which the content parameter is expressed
encoding	the specific encoding of the content field
ontology	the ontology(s) used to give a meaning to the symbols in the content expression
protocol	the interaction protocol that the sending agent is employing with this ACL message
conversation-id	an expression used to identify the sequence of communicative acts that form a conversation
reply-with	an expression that will be used by the responding agent to identify this message
in-reply-to	an expression that references an earlier action to which this message is a reply
reply-by	a time and/or date expression indicating the latest time the sending agent wants a reply

See FIPA standard SC00061G

FIPA Performatives (informally)

accept-proposal	accepting a previous proposal
agree	agreeing to perform some action
cancel	inform another agent that the agent no longer need perform some action
call-for-proposal	calling for proposals to perform an action
confirm	informs a given proposition is true
disconfirm	informs a given proposition is false
failure	an action was attempted but failed
Inform	a given proposition is true
inform-if	inform whether or not a proposition is true
inform-ref	inform the object which corresponds to a descriptor
not-understood	did not understand what the receiver just did
propagate	pass a message on
propose	submitting a proposal to perform an action
proxy	pass on an embedded message
query-if	asking whether or not a proposition is true
query-ref	asking for the object referred to
refuse	refusing to perform an action
reject-proposal	rejecting a proposal during negotiation
request	request to perform some action
request-when	request to perform some action when some proposition becomes true
request-whenever	request to perform some action each time the proposition becomes true
subscribe	requesting to notify of the value of a reference whenever the object changes

See FIPA standard SC00037J

FIPA Semantics

Agent

Semantic content

Proposition

<i, inform (j, φ)>

Feasibility Precondition

FP: $B_i\varphi \wedge \neg B_i(Bif_j\varphi \vee Uif_j\varphi)$

RE: $B_j\varphi$

Rational Effect

"j believes φ"

$B_j\varphi \vee B_j\neg\varphi$

$U_j\varphi \vee U_j\neg\varphi$

(where U_j means "j is uncertain about φ")

FIPA *inform* semantics

- Formally:

  <i, inform (j, φ)>

  FP: $B_i\varphi \wedge \neg B_i(Bif_j\varphi \vee Uif_j\varphi)$

  RE: $B_j\varphi$

- Example:

```
(inform
    :sender (agent-identifier :name i)
    :receiver (set (agent-identifier :name j))
    :content "weather (today, raining)"
    :language Prolog)
```


FIPA *request* semantics

- Formally:

  <i, request (j, *a*)>

  FP: FP(*a*) [i\j] $\wedge$ B_i Agent (j, *a*) $\wedge$ $\neg B_i$ I_j Done (*a*)

  RE: Done (a)

  FP(*a*) [i\j] denotes the part of the FPs of *a* which are mental attitudes of i.

- Example:

```
(request
    :sender (agent-identifier :name i)
    :receiver (set (agent-identifier :name j))
    :content "open \"db.txt\" for input"
    :language vb)
```