Interaction and Communication

Agent-Oriented Software Engineering

A.A. 2019-2020

Prof. Paolo Giorgini

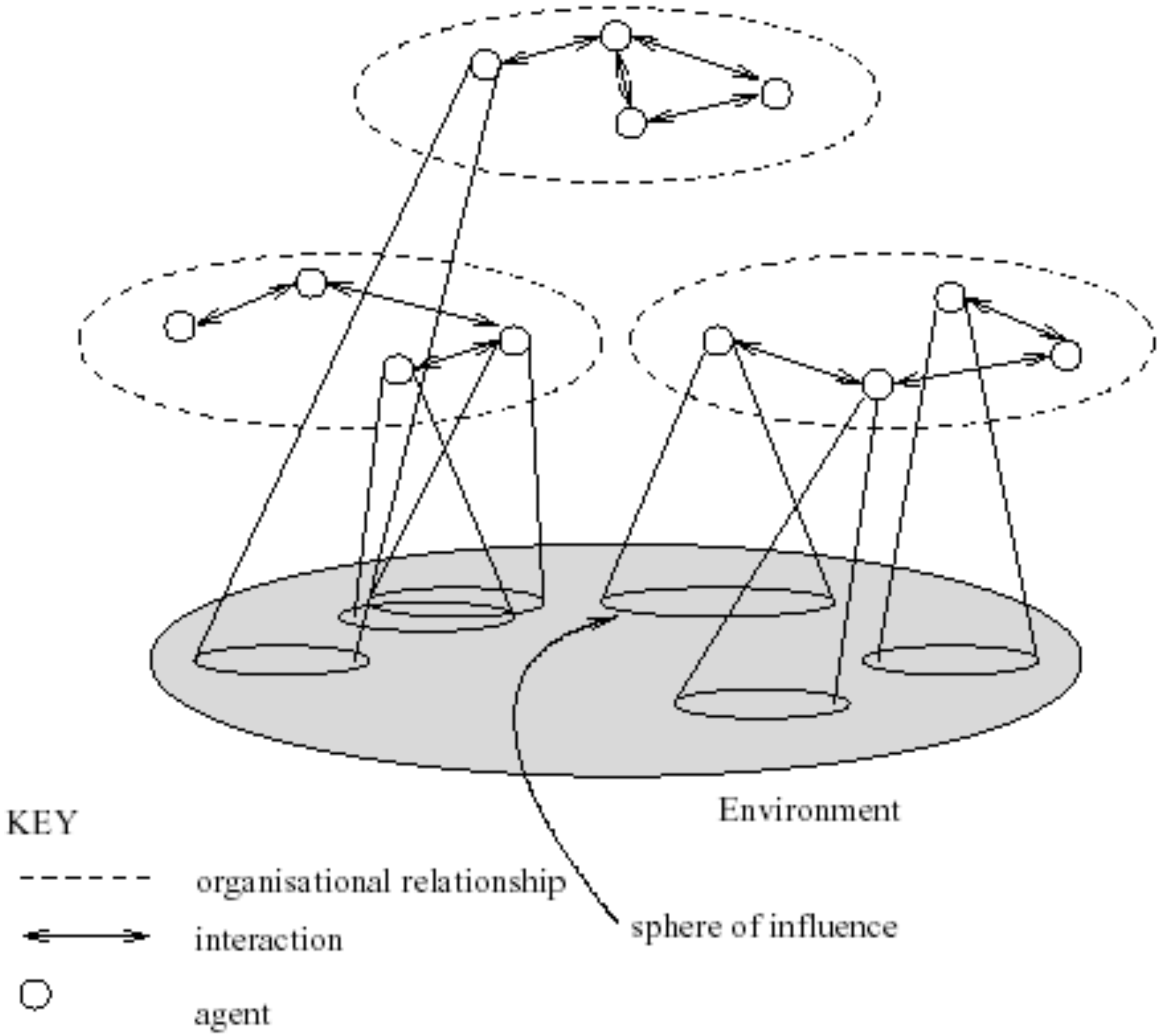
UNIVERSITY OF TRENTO - Italy

Department of Information
and Communication Technology

Agents interact one another

A multiagent system contains a number of agents…

- …which interact through communication…
- …are able to act in an environment…
- …have different "spheres of influence" (which may coincide)…
- …will be linked by other (organizational) relationships`



Utilities and Preferences

- Assume we have just two agents: $Ag = \{i, j\}$
- Agents are assumed to be *self-interested*:
  - they *have preferences over how the environment is*
- Assume $\Omega = \{\omega_1, \omega_2, \ldots\}$ is the set of "outcomes" that agents have preferences over
- We capture preferences by *utility functions*:

$$u_i = \Omega \rightarrow \mathrm{R}$$
$$u_j = \Omega \rightarrow \mathrm{R}$$

- Utility functions lead to *preference orderings* over outcomes:

$$\omega >_i \omega' \text{ means } u_i(\omega) > u_i(\omega')$$
$$\omega \geq_i \omega' \text{ means } u_i(\omega) \geq u_i(\omega')$$

Utility-based agent

- A measure of how desirable a particular state (goal) is.
- *Utility function* maps a state to a measure of the utility of the state
  - Can describe how "happy" the agent is
- A rational utility-based agent chooses the action that maximizes the expected utility of the action outcomes
  - that is, the agent expects to derive, on average, given the probabilities and utilities of each outcome.

Multiagent Encounters

- We need a model of the environment in which these agents will act…
  - agents simultaneously choose an action to perform, and as a result of the actions they select, an outcome in Ω will result
  - the *actual* outcome depends on the *combination* of actions
  - assume each agent has just two possible actions that it can perform, *C* ("Cooperate") and *D* ("Defect")
- Environment behavior given by *state transformer function*:

$$\tau : \underbrace{Ac}_{\text{agent } i\text{'s action}} \times \underbrace{Ac}_{\text{agent } j\text{'s action}} \to \Omega$$

Multiagent Encounters

- Here is a state transformer function:

  $$\tau(D,D)=\omega_1 \quad \tau(D,C)=\omega_2 \quad \tau(C,D)=\omega_3 \quad \tau(C,C)=\omega_4$$

  (This environment is sensitive to actions of both agents.)

- Here is another:

  $$\tau(D,D)=\omega_1 \quad \tau(D,C)=\omega_1 \quad \tau(C,D)=\omega_1 \quad \tau(C,C)=\omega_1$$

  (Neither agent has any influence in this environment.)

- And here is another:

  $$\tau(D,D)=\omega_1 \quad \tau(D,C)=\omega_2 \quad \tau(C,D)=\omega_1 \quad \tau(C,C)=\omega_2$$

  (This environment is controlled by *j*.)

Rational Action

- Suppose we have the case where *both* agents can influence the outcome, and they have utility functions as follows:

$$u_i(\omega_1) = 1 \quad u_i(\omega_2) = 1 \quad u_i(\omega_3) = 4 \quad u_i(\omega_4) = 4$$
$$u_j(\omega_1) = 1 \quad u_j(\omega_2) = 4 \quad u_j(\omega_3) = 1 \quad u_j(\omega_4) = 4$$

- With a bit of abuse of notation:

$$u_i(D, D) = 1 \quad u_i(D, C) = 1 \quad u_i(C, D) = 4 \quad u_i(C, C) = 4$$
$$u_j(D, D) = 1 \quad u_j(D, C) = 4 \quad u_j(C, D) = 1 \quad u_j(C, C) = 4$$

- Then agent i' s preferences are:

$$C, C \succeq_i C, D \quad \succ_i \quad D, C \succeq_i D, D$$

- "*C*" is the *rational choice* for i.
  (Because i prefers all outcomes that arise through *C* over all outcomes that arise through *D*.)

Payoff Matrices

- We can characterize the previous scenario in a *payoff matrix*:

		i	
		defect	coop
j	defect	1 1	4 1
	coop	1 4	4 4

- Agent i is the *column player*
- Agent j is the *row player*

Dominant Strategies

- Given any particular strategy s (either *C* or *D*) of agent i, there will be a number of possible outcomes
- We say s_1 *dominates* s_2 if every outcome possible by i playing s_1 is preferred over every outcome possible by i playing s_2
- A rational agent will never play a dominated strategy
- So in deciding what to do, we can *delete dominated strategies*
- Unfortunately, there isn't always a unique undominated strategy

Nash Equilibrium

- In general, we will say that two strategies s_1 *and* s_2 are in Nash equilibrium if:
  - under the assumption that agent i plays s_1, agent j can do no better than play s_2; and
  - under the assumption that agent j plays s_2, agent i can do no better than play s_1.
- *Neither agent has any incentive to deviate from a Nash equilibrium*
- Unfortunately:
  - *Not every interaction scenario has a Nash equilibrium*
  - *Some interaction scenarios have more than one Nash equilibrium*

Competitive and Zero-Sum Interactions

- Where preferences of agents are diametrically opposed we have *strictly competitive* scenarios
- Zero-sum encounters are those where utilities sum to zero:

  $$u_i(\omega) + u_j(\omega) = 0 \quad \text{for all } \omega \in \Omega$$

- Zero sum implies strictly competitive
- Zero sum encounters in real life are very rare … but people tend to act in many scenarios as if they were zero sum

The Prisoner's Dilemma

- Two men are collectively charged with a crime and held in separate cells, with no way of meeting or communicating. They are told that:
  - if one confesses and the other does not, the confessor will be freed, and the other will be jailed for three years
  - if both confess, then each will be jailed for two years
  - Both prisoners know that if neither confesses, then they will each be jailed for 1 year

The Prisoner's Dilemma

i

j

	Defect	Cooperate
Defect	3 (1 year) 3 (1 year)	4 (freed) 1 (3 years)
Cooperate	1 (3 years) 4 (freed)	2 (2 years) 2 (2 years)

The Prisoner’ s Dilemma

- Best strategy is (C,C)
  - Confess ---> from 0 to 2 years
  - Defect ---> from 1 to 3 years
- The strategy defect is dominated by confess
- Eliminating the dominated strategies we have that both prisoners confess (and both get 2 years)
- However the best for each of them is to defect (1 year)
  - Suppose they both agree before being arrested to defect
    - If now one of them does not respect the agreement he will be freed
    - So what to do: confess or not confess?

The Prisoner’s Dilemma

- This apparent paradox is *the fundamental problem of multi-agent interactions*
  - *cooperation will not occur in societies of self-interested agents*
- Real world examples:
  - nuclear arms reduction (“why don’t I keep mine. . . ”)
- The prisoner’s dilemma is *ubiquitous*

Working Together

- Why and how do agents work together?
- Important to make a distinction between:
  - *benevolent agents*
  - *self-interested agents*

Benevolent Agents

- If we "own" the whole system, we can design agents to help each other whenever asked
- In this case, we can assume agents are *benevolent*: our best interest is their best interest
- Problem-solving in benevolent systems is *cooperative distributed problem solving* (CDPS)
- *Benevolence simplifies the system design task enormously!*

Self-Interested Agents

- If agents represent individuals or organizations, (the more general case), then we cannot make the benevolence assumption
- Agents will be assumed to act to further their own interests, possibly at expense of others
- Potential for *conflict*
- May complicate the design task enormously

Task Sharing and Result Sharing

- Two main modes of cooperative problem solving:
  - *task sharing*:
    components of a task are distributed to component agents
  - *result sharing*:
    information (partial results, etc.) is distributed

The Contract Net

- A well known task-sharing protocol for *task allocation* is the *contract net*:
  - Recognition
  - Announcement
  - Bidding
  - Awarding
  - Expediting

Recognition

- In this stage, an agent recognizes it has a problem it wants help with.
  Agent has a goal, and either…
  - realizes it cannot achieve the goal in isolation — does not have capability
  - realizes it would prefer not to achieve the goal in isolation (typically because of solution quality, deadline, etc.)

Announcement

- In this stage, the agent with the task sends out an *announcement* of the task which includes a *specification* of the task to be achieved
- Specification must encode:
  - description of task itself (maybe executable)
  - any constraints (e.g., deadlines, quality constraints)
  - meta-task information (e.g., "bids must be submitted by…")
- The announcement is then *broadcast*

Bidding

- Agents that receive the announcement decide for themselves whether they wish to *bid* for the task
- Factors:
  - agent must decide whether it is capable of expediting task
  - agent must determine quality constraints & price information (if relevant)
- If they do choose to bid, then they submit a *tender*

Awarding & Expediting

- Agent that sent task announcement must choose between bids & decide who to "award the contract" to
- The result of this process is communicated to agents that submitted a bid
- The successful *contractor* then expedites the task
- May involve generating further manager-contractor relationships: *sub-contracting*

Efficiency Modifications

- Focused addressing — when general broadcast isn't required
- Directed contracts — when manager already knows which node is appropriate
- Request-response mechanism — for simple transfer of information without overhead of contracting
- Node-available message — reverses initiative of negotiation process

Example: Signal Task Announcement

To: *
From: 25
Type: Task Announcement
Contract: 22–3–1
Eligibility Specification:
Must-Have SENSOR
Must-Have Position Area A
Task Abstraction:
Task Type Signal
Position LAT 47N LONG 17E
Area Name A Specification (…)
Bid Specification: Position Lat Long
Every Sensor Name Type
Expiration Time: 28 1730Z APR 2017

Example: Signal Bid

To: 25
From: 42
Type: BID
Contract: 22–3–1
Node Abstraction:
LAT 47N LONG 17E
Sensor Name S1 Type S
Sensor Name S2 Type S
Sensor Name T1 Type T

Example: Signal Award

To: 42
From: 25
Type: AWARD
Contract: 22–3–1
Task Specification:
Sensor Name S1 Type S
Sensor Name S2 Type S

Agent Communication

- *Macro-aspects* of intelligent agent technology: those issues relating to the agent *society*, rather than the individual:
  - *communication;*
    speech acts; KQML & KIF; FIPA ACL
  - *cooperation*

Speech Acts

- Most treatments of communication in (multi-) agent systems borrow their inspiration from *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**
- The origin of speech act theories are usually traced to Austin's 1962 book, *How to Do Things with Words*

Speech Acts

- 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
- A theory of how utterances are used to achieve intentions is a speech act theory

Different Aspects of Speech Acts

- From "A Dictionary of Philosophical Terms and Names"
- "*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."

Different Aspects of Speech Acts

- “*Illocutionary act:* the speech act of doing something else – offering advice, 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

Different Aspects of Speech Acts

- "*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

Speech Acts

- Searle ('69) identified 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'
  - *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

Speech Acts

- 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")

Speech Acts

- Consider:
  - 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?”

Plan Based Semantics

- 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*

Plan-Based Semantics

- Here is their semantics for *request*: $request(s, h, \phi)$

pre:

- s believe h can do ϕ
  (you don't ask someone to do something unless you think they can do it)
- s believe h believe h can do ϕ
  (you don't ask someone unless *they* believe they can do it)
- s believe s want ϕ
  (you don't ask someone unless you want it!)

post:

- h believe s believe s want ϕ
  (the effect is to make them aware of your desire)

What about inform?

We could adopt that: `inform(s,h,ϕ)`

Pre:
`s` believes ϕ is true ^
`s` intends that `h` believes ϕ

Post:
`s` believes that `h` believes ϕ ^
`h` believes ϕ

Other semantics of course can be adopted

KQML and KIF

- We now consider *agent communication languages* (ACLs) — standard formats for the exchange of messages
- The best known ACL is KQML, developed by the ARPA knowledge sharing initiative
  KQML is comprised of two parts:
  - the Knowledge Query and Manipulation Language (KQML)
  - the Knowledge Interchange Format (KIF)

KQML and KIF

- KQML is a language, that defines various acceptable 'communicative verbs', or *performatives*
  Example performatives:
  - `ask-if` ('is it true that. . . ')
  - `perform` ('please perform the following action. . . ')
  - `tell` ('it is true that. . . ')
  - `reply` ('the answer is . . . ')
- KIF is a language for expressing message *content*

KQML performatives

Category	Reserved performative names
Basic informational performatives	tell, deny, untell, cancel,
Basic query performatives	evaluate, reply, ask-if, ask-about, ask-one, ask-all, sorry
Multi-response query performatives	stream-about, stream-all
Basic effector performatives	achieve, □nachieved
Generator performatives	standby, ready, next, rest, discard, generator
Capability definition performatives	advertise
Notification performatives	subscribe, monitor
Networking performatives	register, unregister, forward, broadcast, pipe, break
Facilitation performatives	broker-one, broker-all, recommend-one, recommend-all, recruit-one, recruit-all

KIF – Knowledge Interchange Format

Used to state:

- Properties of things in a domain (e.g., "`Paolo is chairman`")
- Relationships between things in a domain (e.g., "`Paolo is the Trento's boss`")
- General properties of a domain (e.g., "`All students are registered for at least one course`")

KIF – Knowledge Interchange Format

- “The temperature of m1 is 83 Celsius”:

```
(= (temperature m1) (scalar 83 Celsius))
```

- “An object is a bachelor if the object is a man and is not married”:

```
(defrelation bachelor (?x) :=
     (and (man ?x) (not (married ?x))))
```

- “Any individual with the property of being a person also has the property of being a mammal”:

```
(defrelation person (?x) :=> (mammal ?x))
```

KQML and KIF

- In order to be able to communicate, agents must have agreed on a common set of terms
- A formal specification of a set of terms is known as an *ontology*
- The knowledge sharing effort has associated with it a large effort at defining common ontologies — software tools like ontolingua for this purpose
- Example KQML/KIF dialogue...

```
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))
```

KQML examples

A query about the price of a share of IBM stock

```
(ask-one
    :content (PRICE IBM ?price)
    :receiver stock-server
    :language LPROLOG
    :ontology NYSE-TICKS)

(ask-all
     :content "price(IBM, [?price, ?time])"
     :receiver stock-server
     :language standard_prolog
     :ontology NYSE-TICKS)
```

FIPA

- The Foundation for Intelligent Physical Agents (IEEE- FIPA standard) —> the FIPA ACL
- Basic structure is quite similar to KQML:
  - *performative*
    20 performative in FIPA
  - *housekeeping*
    e.g., sender, etc.
  - *content*
    the actual content of the message

FIPA

- Example:

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

FIPA-ACL performative set

Accept proposal	Inform If	Refuse
Agree	Inform Ref	Reject Proposal
Cancel	Not Understood	Request
Call for Proposal	Propagate	Request When
Confirm	Propose	Request Whenever
Disconfirm	Proxy	Subscribe
Failure	Query If	
Inform	Query Ref	

FIPA

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			

FIPA-ACL message elements

Element	Description
Performative	Denotes the type of the communicative act of the ACL message
Sender	Denotes the identity of the sender (the name of the agent) of the message
Receiver	Denotes the identity of the intended recipient of the message
Reply-To	This element indicates that subsequent messages in this conversation thread are to be directed to the agent named in the reply-to-element, instead of the agent named in the sender element
Content	Denotes the content of the message
Language	Denotes the language in which the content element is expressed
Encoding	Denotes the specific encoding of the content language expression
Ontology	Denotes the ontology(s) used to give a meaning to the symbols in the content expression
Protocol	Denotes the interaction protocol that the sending agent is employing with this ACL message
Conversation-id	Introduces an expression (a conversation identifier) which is used to identify the ongoing sequence of communicative acts that together form a conversation
Reply-with	Introduces an expression that will be used by the responding agent to identify this message
In-reply-to	Denotes an expression that references an earlier action to which this message is a reply
Reply-By	Denotes a time and/or date expression which indicates the latest time by which the sending agent would like to have received a reply

“Inform” and “Request”

- “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:
    - pre-condition
      what must be true in order for the speech act to succeed
    - “rational effect”
      what the sender of the message hopes to bring about

“Inform” and “Request”

- For the “inform” performative…
- The content is a *statement*.
- Pre-condition is that sender:
  - holds that the content is true
  - intends that the recipient believe the content
  - does not already believe that the recipient is aware of whether content is true or not

“Inform” and “Request”

- For the “request” performative…
- The content is an *action*.
- Pre-condition is that sender:
  - intends action content to be performed
  - believes recipient is capable of performing this action
  - does not believe that receiver already intends to perform action

An example of FIPA-ACL

```
(cfp
    :sender (agent-identifier :name BuyerAgent)
    :receiver (set (agent-identifier :name
     SellerAgent))
    :content
      ((action (agent-identifier :name SellerAgent)
      (sell :movie Gladiator))
      (any ?x (and (= ?x (price Gladiator)) (< ?x 20))))
    :ontology movie-ontology
    :language FIPA-SL)
```

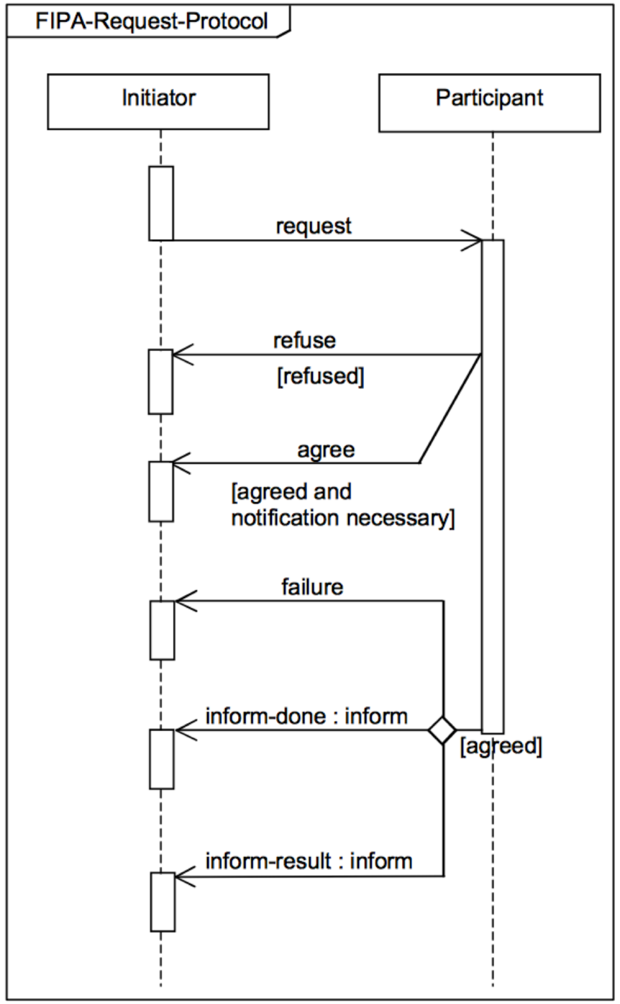
FIPA Semantic Language

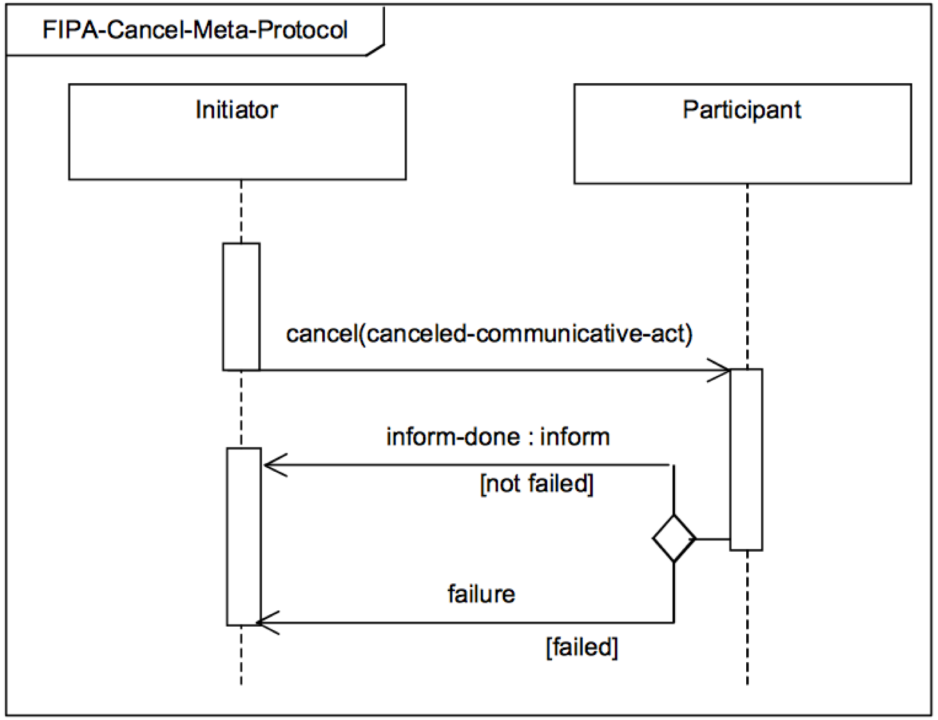
- FIPA Semantic Language (FIPA SL) is a formal language to define the content of the FIPA-ACL.
- FIPA SL can be used to express objects, propositions and actions.
    - Object expression is used to declare variables and make assertions.
    - Action expressions describe some action that is either already performed, intended to be performed in the future or is currently being performed.
    - Propositions are used to represent the behavioural aspects of agents like goals, intents, beliefs and uncertainty.
- For example, agents may have persistent goals stated in the form

```
(PG < agent > < expression >)
```

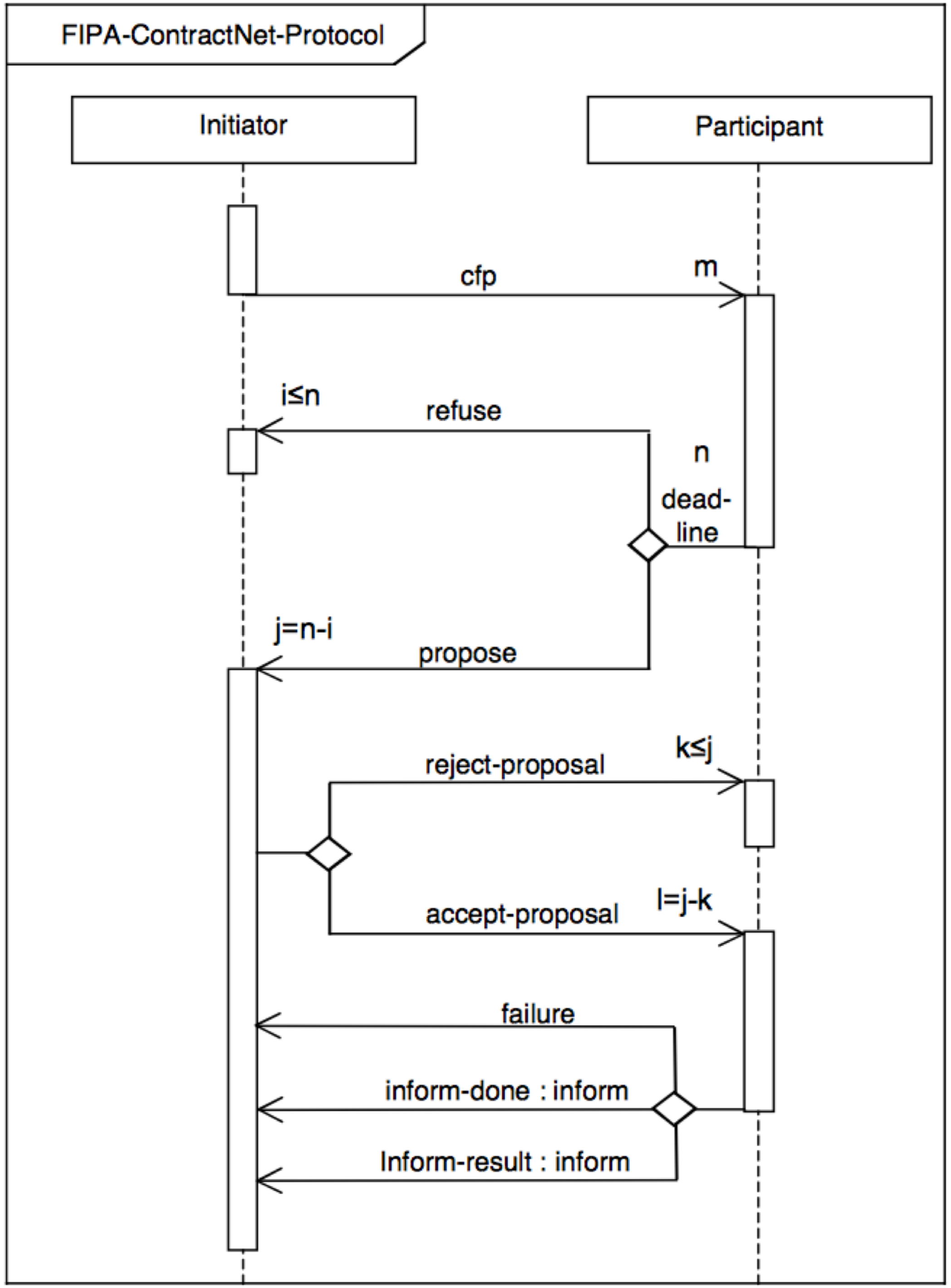
- This states that an agent holds a persistent goal that expression becomes true but will not necessarily possess a plan to achieve this.

FIPA REQUEST protocol





FIPA ContractNet Protocol



Exercise

Personal agents represent their users (‘humans’) in the organization of football (‘calcetto’) match.

- `Starter` agent has the initiative to organize the match
- `Participant` agent checking the user’s agenda will decide either to participate or not
- `Time` and `location` can be negotiated
- `Deadline` is fixed by the `Starter`