Lecture #3

Damavand university

Communication in multiagent systems

Agent communication

- Artificial communication
- Indirect communication
- Direct communication
- ACL
- Communication protocols

1. Artificial communication

Human communication

- Communication is the intentional exchange of information brought about by the production and perception of signs drawn from a shared system of conventional signs (AIMA, Russell&Norvig) ➔ ***language***
- Communication seen as an **action** (communicative act) and as an **intentional stance**

Component steps of communication

Speaker
- ⇦ Intention
- ⇦ Generation
- ⇦ Synthesis

Hearer
- ⇨ Perception
- ⇨ Analysis
- ⇨ Disambiguation
- ⇨ Incorporation

→ **Syntax**, **Semantics**, **Pragmatics**

Agent communication

- low-level communication: simple signals, traces, low-level languages
- high-level communication - cognitive agents, mostly seen as intentional systems
- Communication in MAS = more than simple communication, implies **interaction**

Communication protocols = enables agents to exchange and understand messages

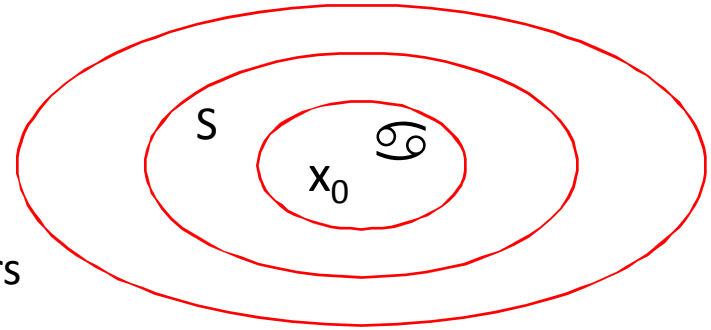
Interaction protocols = enable agents to have **conversations**, i.e., structured exchanges of messages

2. Indirect communication

2.1 Signal propagation - Manta, A. Drogoul

- An agent sends a signal, which is broadcast into the environment, and whose intensity decreases as the distance decreases
- At a point *x*, the signal may have one of the following intensities

$V(x)=V(x_0)/dist(x,x_0)$ $V(x)=V(x_0)/dist(x,x_0)^2$



S - stimulus

Agent A
(stimulus triggers behavior P)

➔Topological differences lead to social differences

Agent B
(stimulus triggers behavior P)

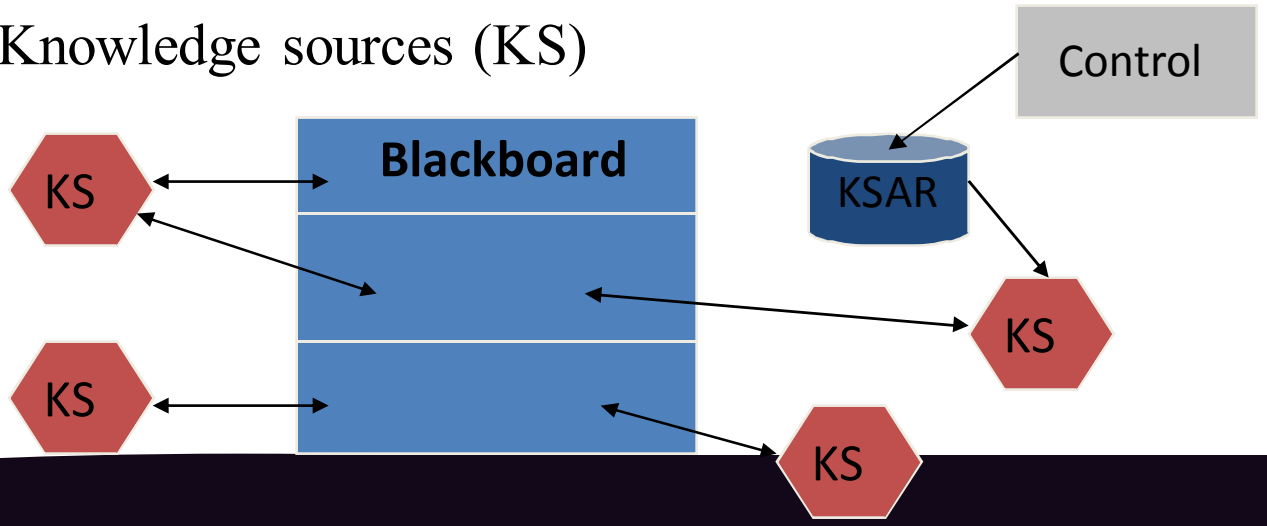
2.2 Trails - L. Steels

agents drop "radioactive crumbs" making trails

an agent following a trail makes the trail faint until it disappears

2.3 Blackboard systems, Barbara Hayes-Roth, 1985

- Blackboard = a common area (shared memory) in which agents can exchange information, data, knowledge
- Agents initiates communication by writing info on the blackboard
- Agents are looking for new info, they may filter it
- Agents must register with a central site to receive an access authorization to the blackboard
- Blackboard = a powerful distributed knowledge computation paradigm
- Agents = Knowledge sources (KS)



3. Direct communication

Sending messages

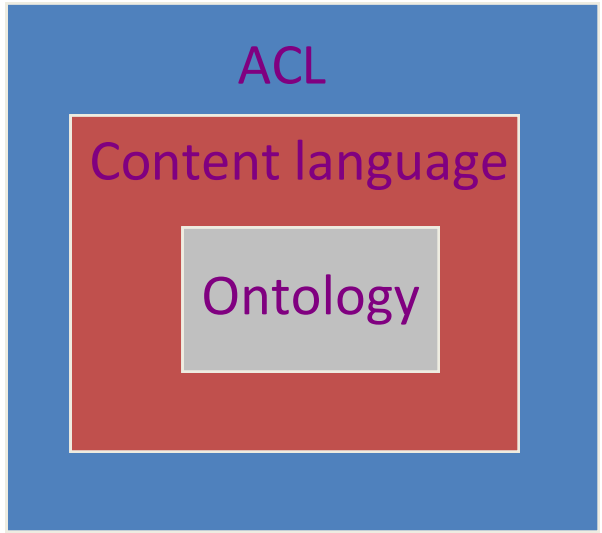
- method invocation – Actors
- exchange of partial plans – coordination of cooperative agents

ACL = Agent Communication Languages

communication as action - **communicative acts**

3.1 Agent Communication Languages

- Concepts (distinguish ACLs from RPC, RMI or CORBA, ORB):
  - ACLs handle propositions, rules, and actions instead of objects with no associated semantics
  - An ACL message describes a desired state in a declarative language, rather than a procedure or method invocation
  - ACLs are mainly based on BDI theories: BDI agents attempt to communicate their BDI states or attempt to alter interlocutor's BDI state
  - ACLs are based on Speech Act Theory
  - Agent behavior and strategy drive communication and lead to conversations



Origins of ACLs

Knowledge Sharing Effort - DARPA, 1990

- External Interface Group - interaction between KBS - KQML
- Interlingua - common language of KB - KIF
- Shared, Reusable Knowledge Bases - Ontolingua

Theory of Speech Acts

J. Austin - How to do things with words, 1962, J. Searle - Speech acts, 1969

A speech act has 3 aspects:

- **locution** = physical utterance by the speaker
- **illocution** = the intended meaning of the utterance by the speaker (performative)
- **prelocution** = the action that results from the locution

Alice told Tom: "**Would you please** close the door"

locution illocution content

prelocution: **door closed** (hopefully!)

Illocutionary aspect - several categories

- **Assertives**, which inform: the door is shut
- **Directives**, which request: shut the door, can pelicans fly?
- **Commissives**, which promise something: I will shut the door
- **Permissive**, which gives permission for an act: you may shut the door
- **Prohibitives**, which ban some act: do not shut the door
- **Declaratives**, which causes events: I name you king of Rom
- **Expressives**, which express emotions and evaluations: I am happy

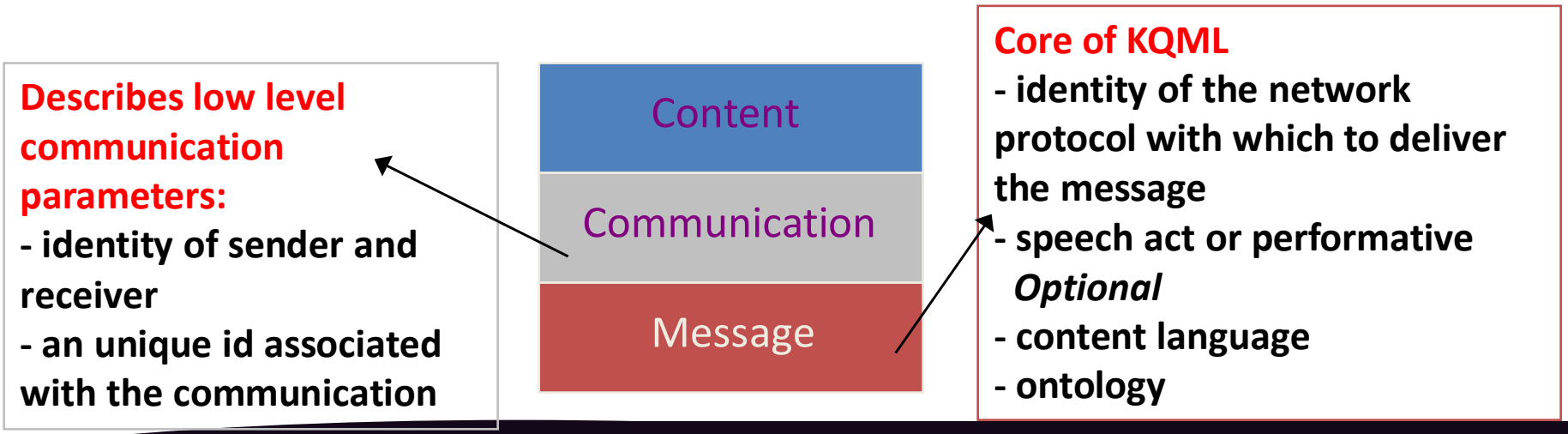
3.2 KQML - Knowledge Query and Manipulation Language

A high-level, message-oriented communication language and protocol for information exchange, independent of content syntax (KIF, SQL, Prolog,…) and application ontology

KQML separates:

- semantics of the communication protocol (domain independent)
- semantics of the message (domain dependent)

3 (conceptual) layers



Syntax

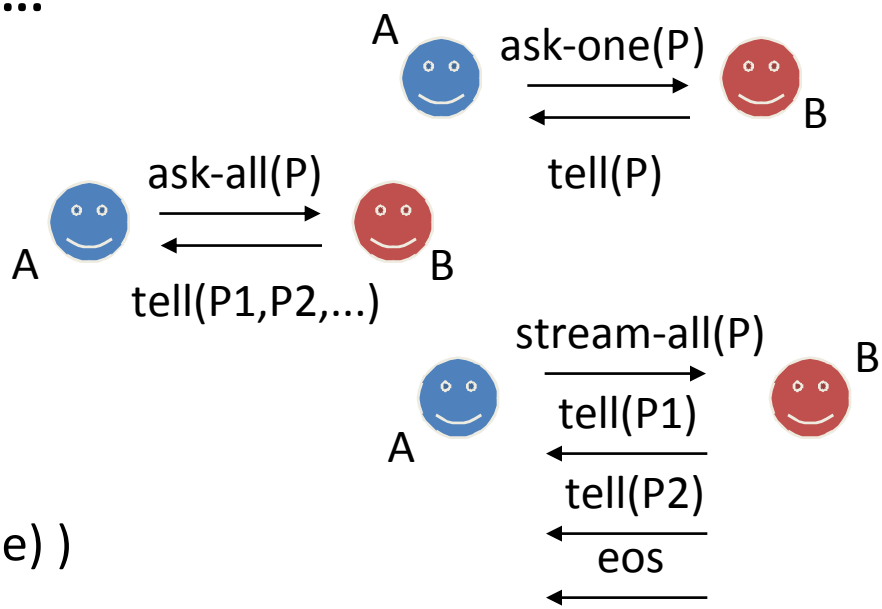
```
(ask-one :sender  joe
         :receiver  ibm-stock
         :reply-with  ibm-stock
         :language  PROLOG
         :ontology NYSE-TICKS
         :content (price  ibm  ?price) )
```

```
(tell :sender  willie
      :receiver  joe
      :reply-with  block1
      :language  KIF
      :ontology BlockWorld
      :content (AND (Block A)(Block B)
                    (On  A  B)) )
```

1. Query performatives:

ask-one, ask-all, ask-if, stream-all,...

```
(stream-all        :sender  willie
                   :receiver  ibm-stock
           :content (price ?VL  ?price  ) )
(standby
           :content (stream-all
                    :content  (price  ?VL  ?price) )
```



2. Generator performatives:

standby, ready, next, rest, discard, generate,...

3. Response performatives:

reply, sorry ...

4. Generic informational performatives:

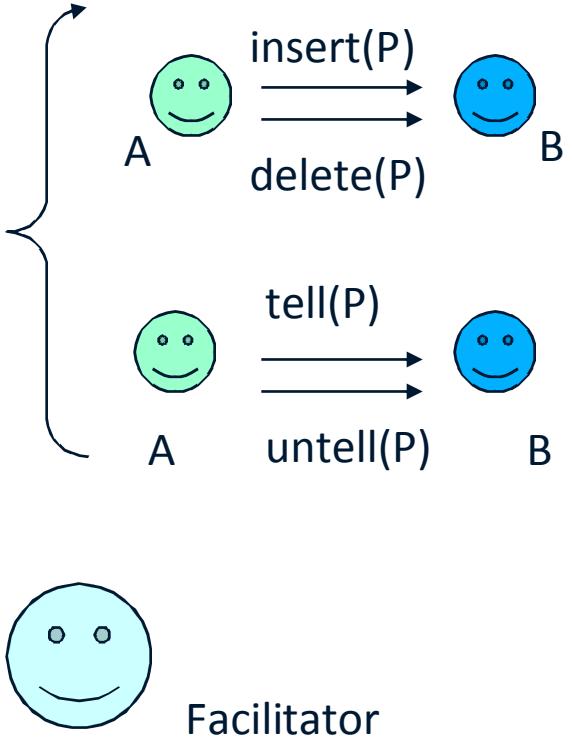
tell, untell, insert, delete, ...

5. Capability performatives:

advartise, subscribe, recommend...

6. Network performatives:

register, unregister, forward, route, ...

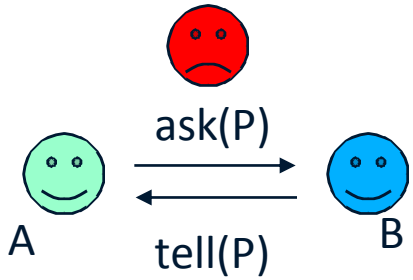


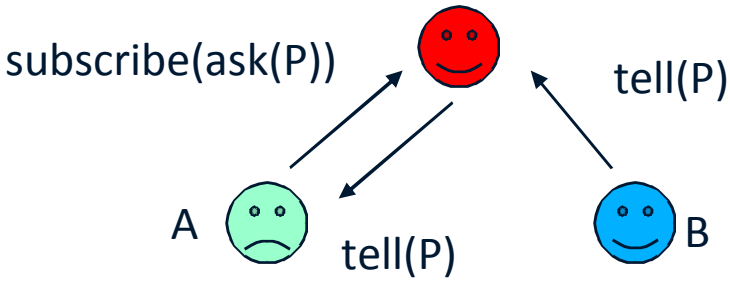
In fact, KQML contains only 2 types of illocutionary acts:
assertives and directives
+ facilitator and network-related performatives (no necessarily speech acts)

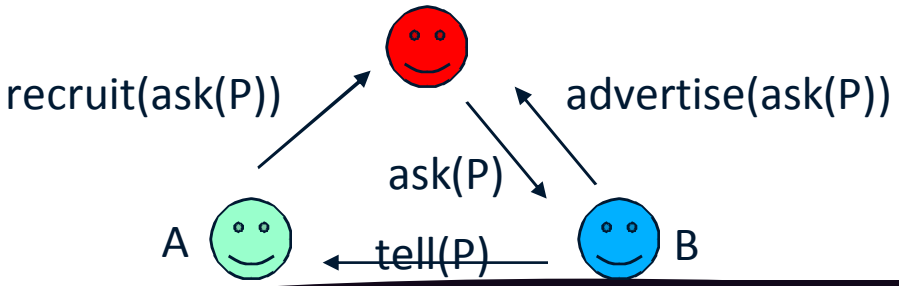
Facilitator agent

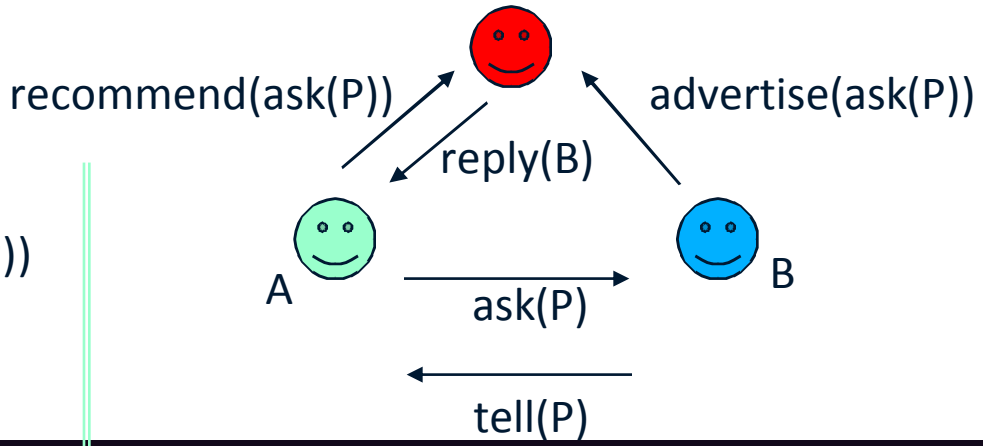
= an agent that performs various useful communication services:

- maintaining a registry of service names (Agent Name Server)
- forwarding messages to named services
- routing messages based on content
- matchmaking between information providers and clients
- providing mediation and translation services









Semantics of KQML (Labrou & Finin)

- Use preconditions and postconditions that govern the use of a performative + the final state for the successful performance of the performative
- Uses propositional attitudes: belief, knowledge, desire, intentions

Preconditions: the necessary states for an agent to send a performative and for the receiver to accept it and successfully process it; if the precondition do not hold, the most likely response is error or sorry

Postconditions - describe the state of the sender after successful utterance of a performative and of the receiver after the receipt and processing of a message

Completion condition - the final state after a conversation has taken place and that the intention associated with the performative that started the conversation has been fulfilled

Propositional attitudes

Bel(A,P) **Know**(A,S) **Want**(A,S) **Int**(A,S)

Instances of action

Proc(A,M) SendMsg(A,B,M)

tell(A,B,X)

A states to B that A believes the content X to be true, Bel(A,X)

Pre(A): Bel(A,X) $\wedge$ Know(A, Want(B, Know(B,S)))

Pre(B): Int(B, Know(B,S))

where S may be any of Bel(B,X) or $\neg$Bel(B,X)

Post(A): Know(A, Know(B, Bel(A,X))) *no unsolicited information*

Post(B): Know(B, Bel(A,X))

Completion: Know(B, Bel(A,X))

advertise(A,B,M)

A states to B that A can and will process the message M from B, if it receives one

Int(A, Proc(A,M)) *commisive act*

Pre(A): Int(Proc(A,M))

Pre(B): NONE

Post(A): Know(A, Know(B, Int(A, Proc(A,M))))

Post(B): Know(B, Int(A, Proc(A,M)))

Completion: Know(B, Int(A, Proc(A,M)))

3.3 FIPA - ACL

Foundation for Intelligent Physical Agents, 1996

- Goal of FIPA = make available specifications that maximize interoperability across agent-based systems
- FIPA Committees: ACL, agent specification, agent-software interaction
- As KQML, FIPA ACL is based on speech act theory; it sees messages as communication acts (CA); syntax similar to KQML
- Differs in: the names of CAs, set of CAs, and semantics

```
(inform     :sender  Agent1
            :receiver  Agent2
            :content (price good2 150)
            :in-reply-to  round-1
            : reply-with bid03
            : language S1
            :ontology hp-auction
            :reply-by  10
            :protocol  offer
            :conversation-id  conv-1 )
```

FIPA - Semantics

SL (***Semantic Language***) - a quantified, multi-modal logic, with modal operators

Allows to represent:

- beliefs
- uncertain beliefs
- desires
- intentions

B φ - belief **D** φ - desire **U** φ - uncertain belief

G φ - intention

Bif φ - express whether an agent has a definite opinion one way or another about the truth or falsity of φ

Uif φ - the agent is uncertain about φ

FIPA - Semantics

The semantics of a CA is specified as a set of SL's formulae that describe:

- **Feasibility preconditions** - the necessary conditions for the sender - the sender is not obliged to perform the CA
- **Rational effect** - the effect that an agent can expect to occur as a result of performing the action; it also typically specifies conditions that should hold true of the recipient
  The receiving agent is not required to ensure that the expected effect comes about
  The sender can not assume that the rational effect will necessary follow

$$\langle i, \text{inform}(j, \varphi)\rangle$$

Pre: $\mathbf{B}_i\,\varphi \wedge \neg \mathbf{B}_i\,(\mathbf{Bif}_j\,\varphi \vee \mathbf{Uif}_j\,\varphi)$

Post: $\mathbf{B}_j\,\varphi$

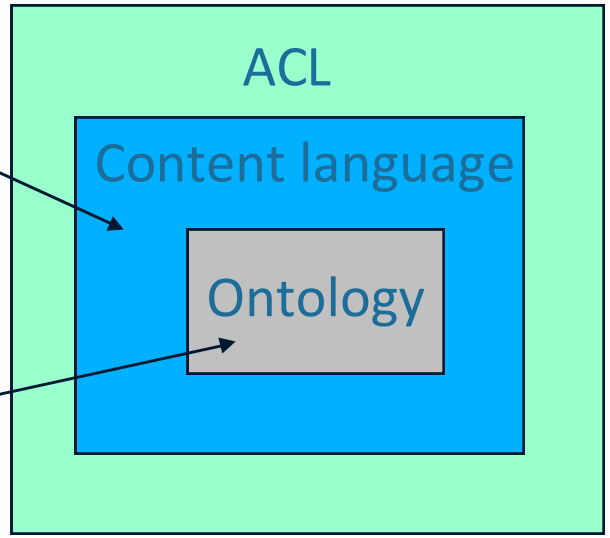
Using ACLs in MAS

Any MAS that is to use an ACL must provide:

- a finite set of APIs for composition, sending, and receiving ACL messages
- an infrastructure of services that assist agents in naming, registration, and basic facilitation services (finding other agents that can do things for your agent)
- code for every reserved message type that takes the action prescribed by the semantics for the particular application;
- the code depends on the application language, the domain, and the details of the agent system using the ACL

4. Communication content

- **Content languages**
  - KIF
  - Prolog
  - Clips
  - SQL
  - FIPA-SL, FIPA-CCL, FIPA-KIF
  - DAML
- **Ontologies**



DARPA Agent Markup Language

The DAML Program officially began in August 2000.
The goal of the DAML effort is to develop a language and tools to facilitate the concept of the Semantic Web.

Content languages

- **KIF - Knowledge Interchange Format**

A prefixed version of FOPL with expressions to support nonmonotonic reasoning and definitions

(salary ?x ?y ?z)

- capability of encoding knowledge about knowledge

(interested joe '(salary ,?x ,?y ,?z))

- **FIPA-SL**

(**request** :sender (agent-identifier :name i)
:receiver (set (agent-identifer :name j)
:content ((action (agent-identifier :name j)
(deliver box7 (loc 10 15))))
:protocol fipa-request
:language fipa-sl
:reply-with order56)

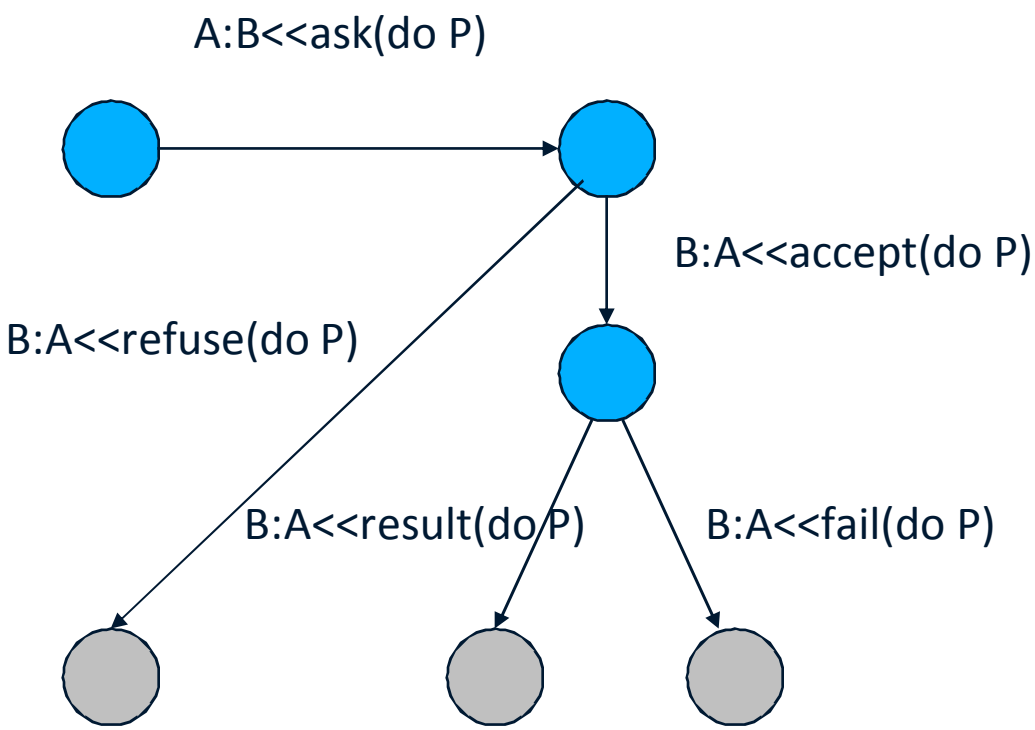
(**agree** :sender (agent-identifier :name j)
:receiver (set (agent-identifer :name i)
:content ((action (agent-identifier :name j)
(deliver box7 (loc 10 15))) (priority order56 low))
:protocol fipa-request
:language fipa-sl
:in-reply-to order56)

5. Interaction protocols

Interaction protocols = enable agents to have **conversations**, i.e., structured exchanges of messages

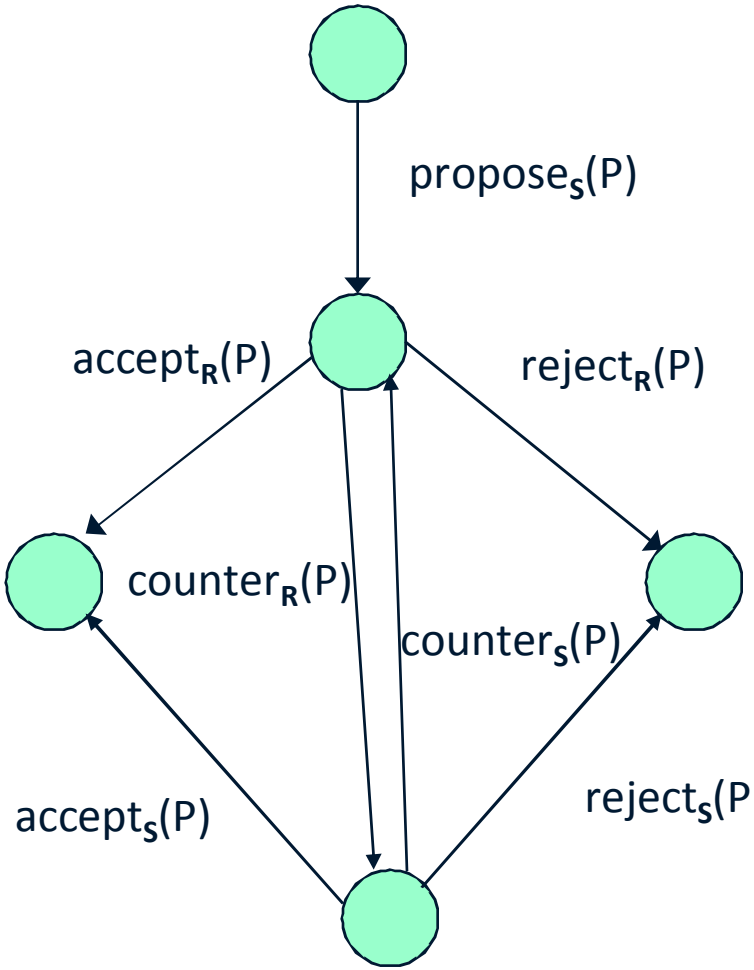
- **Finite automata**
- **Conversations in KQML**
- **Petri nets**
- **FIPA IP standards**:
  - **FIPA-query, FIPA-request, FIPA-contract-net, ...**

5.1 Finite state automata



Winograd, Flores, 1986

COOL, Barbuceanu,95



5.2 Conversations in KQML

Use **Definite Clause Grammars** (DCG) formalism for the specification of conversation policies for KQML performatives

DCGs extend Context Free Grammars in the following way:

- non-terminals may be compound terms
- the body of the rule may contain procedural attachments, written as "{" and "}" that express extra conditions that must be satisfied for the rule to be valid

Ex: noun(N) → [W], {RootForm(W,N), is_noun(N)}

S → **s**(Conv, P, S, R, inR, Rw, IO, Content), {member(P, [advertise, ask-if]}

s(Conv, **ask-if**, S, R, inR, Rw, IO, Content) →

[**ask-if**, S, R, inR, Rw, IO, Content] |

[**ask-if**, S, R, inR, Rw, IO, Content], {OI is inv(IO)},

r(Conv, **ask-if**, S, R, _, Rw, OI, Content)

r(Conv, **ask-if**, R, S, _, inR, IO, Content) →

[**tell**, S, R, inR, Rw, IO, Content] |

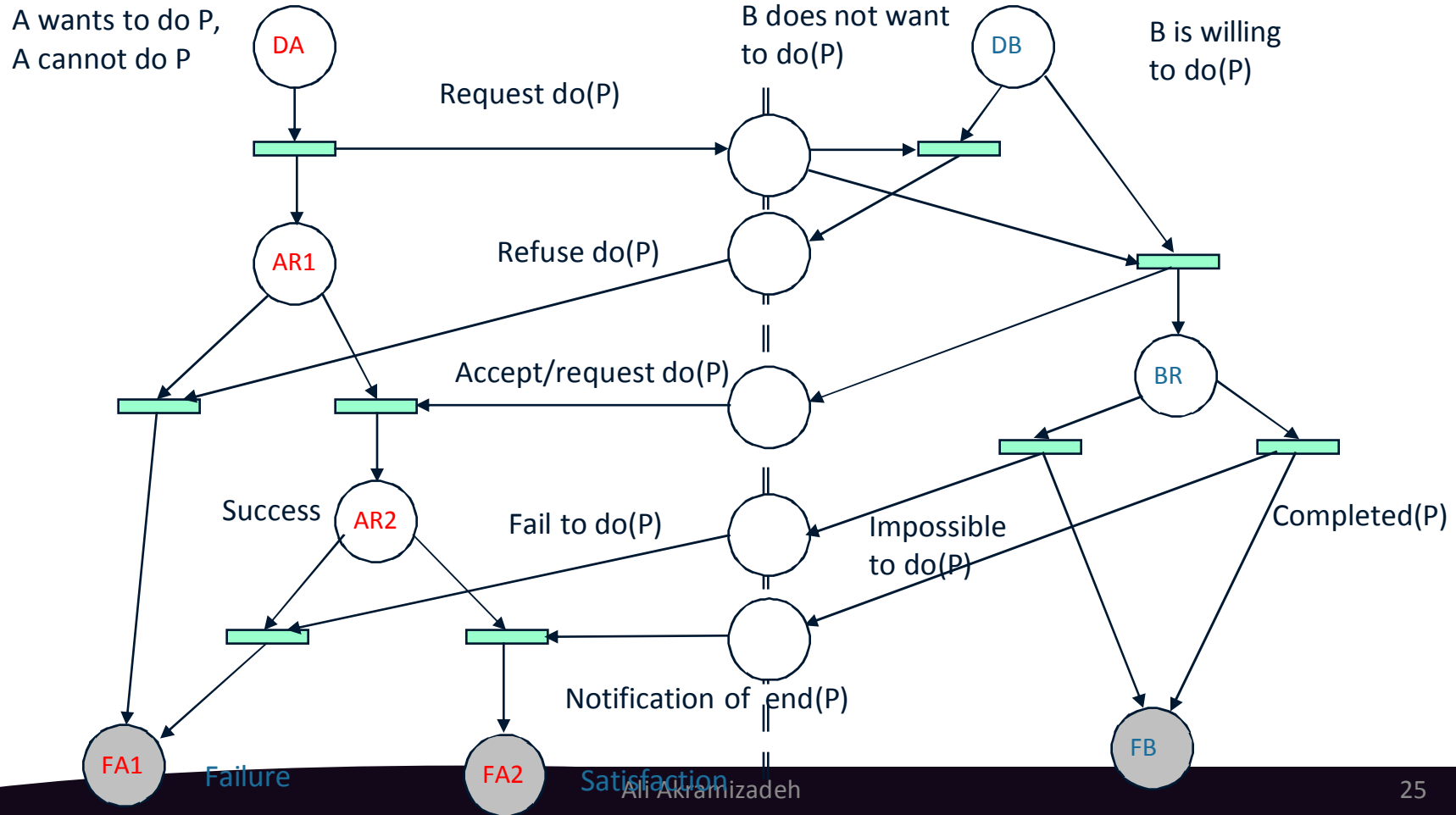
problem(Conv, R, S, inR, _, IO)

Labrou, Finin, 1998

5.3 Petri nets

Petri net = oriented graph with 2 type of nodes: **places** and **transitions**;
there are moving **tokens** through the net - representation of dynamic aspect of processes.
Tokens are moved from place to place, following firing rules.
A transition *T* is enabled if all the input places *P* of *T* posses a token (several other rules may be defined).
A marking is a distribution of tokens over places. Colored Petri-nets

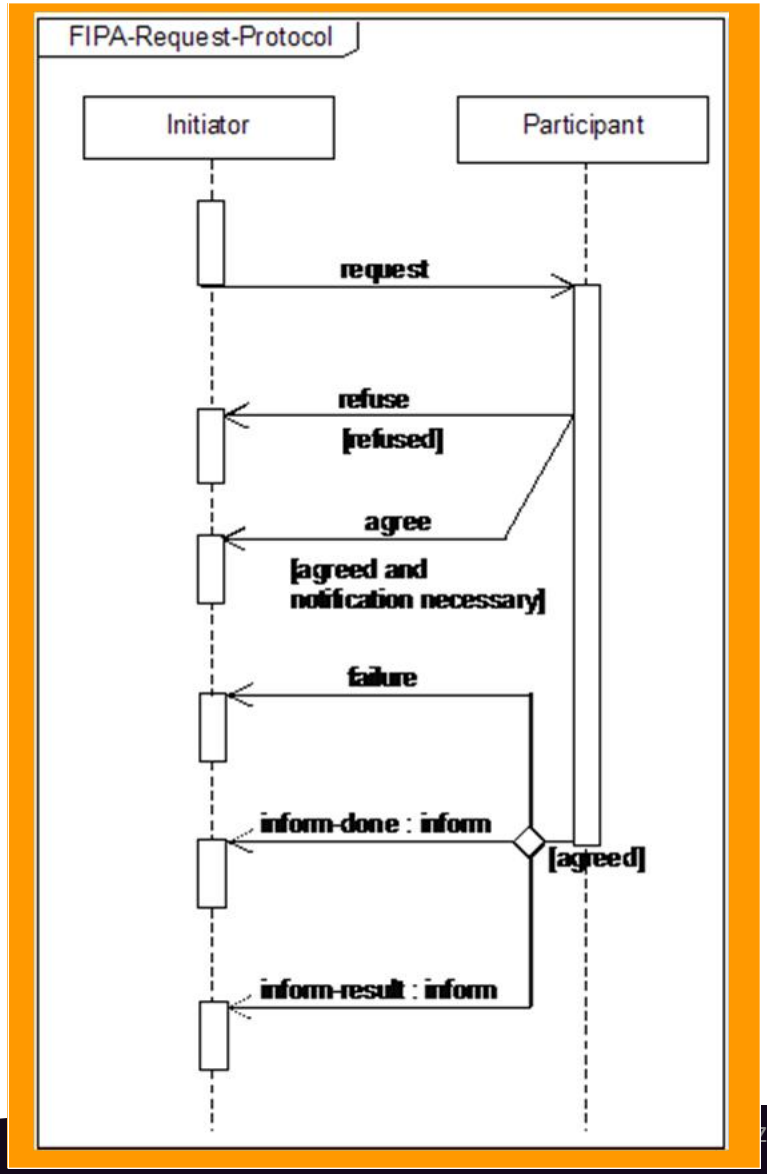
Ferber, 1997

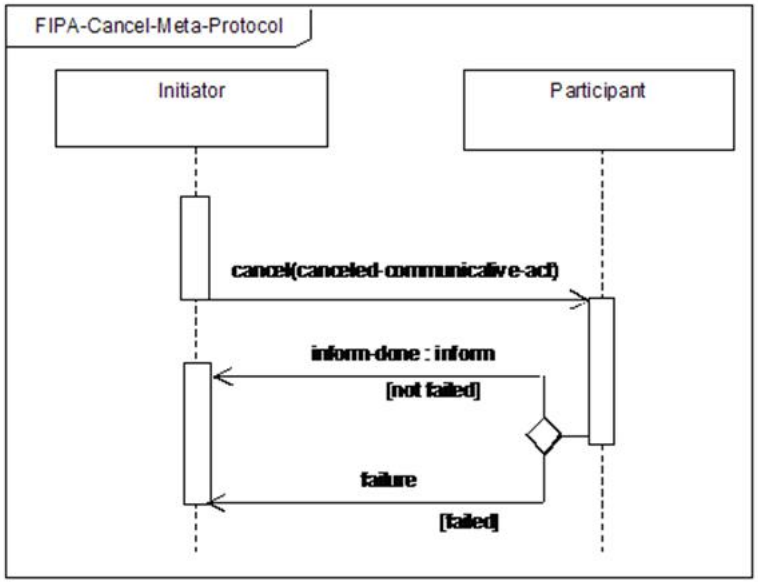


5.4 FIPA – Protocols

- **FIPA-query, FIPA-request, FIPA-contract-net, ...**

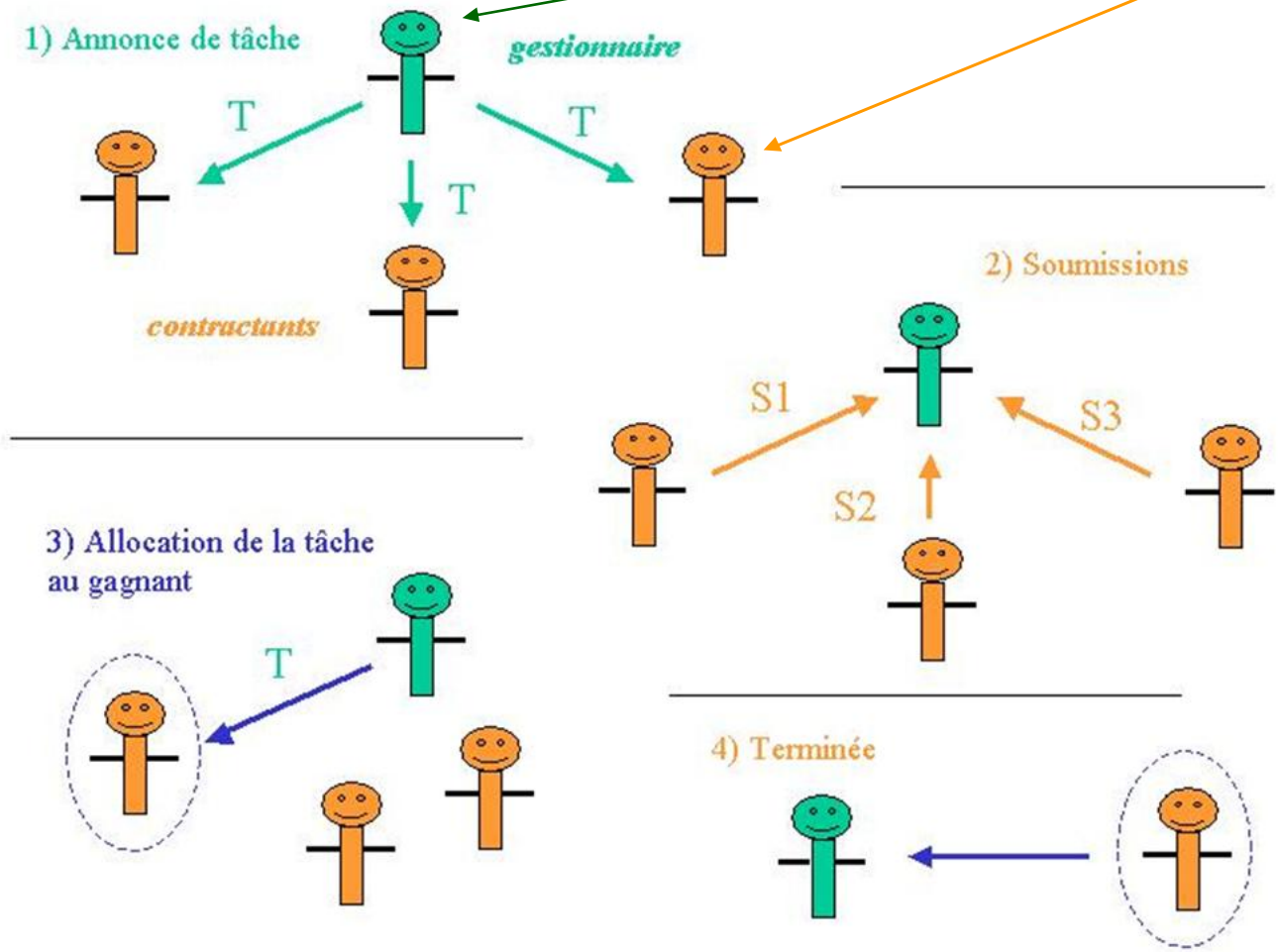
FIPA - Request



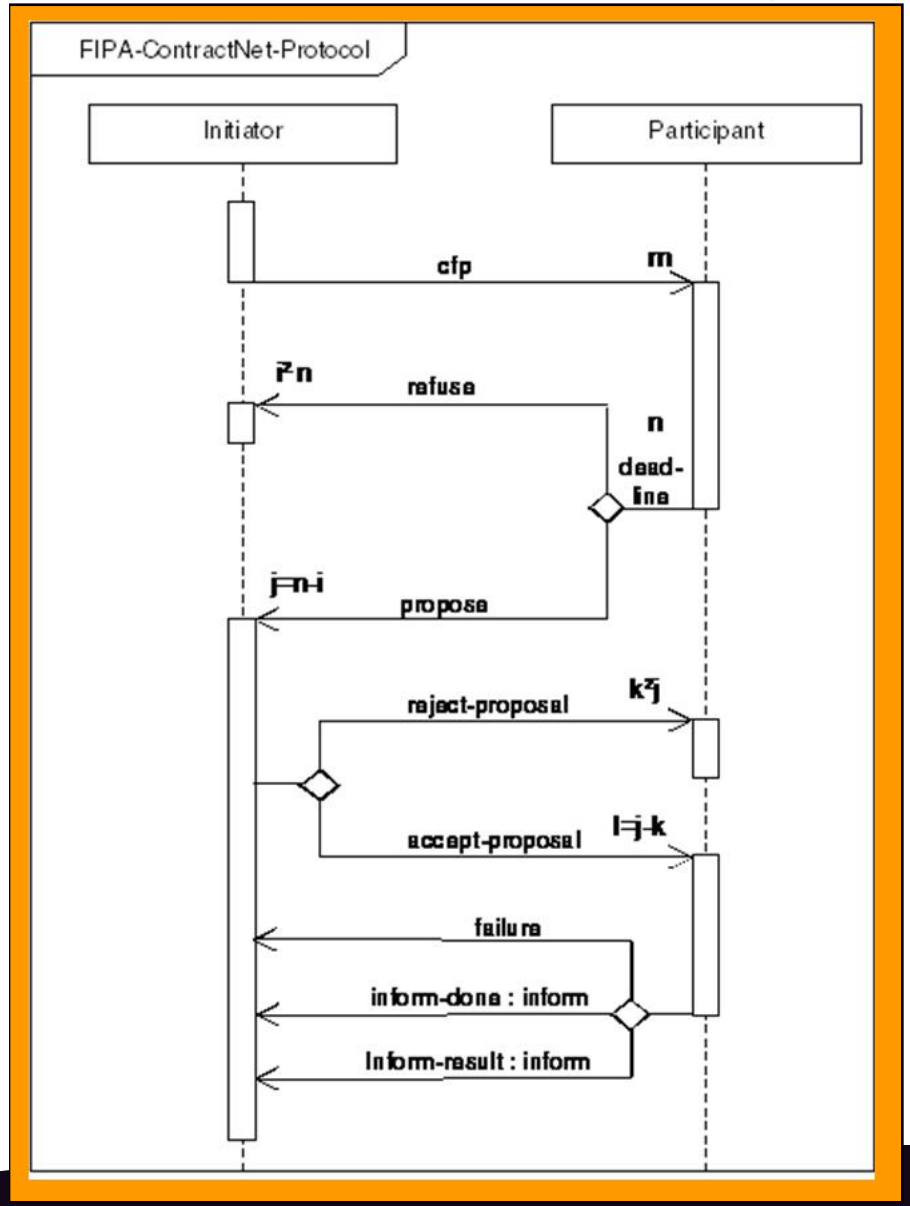


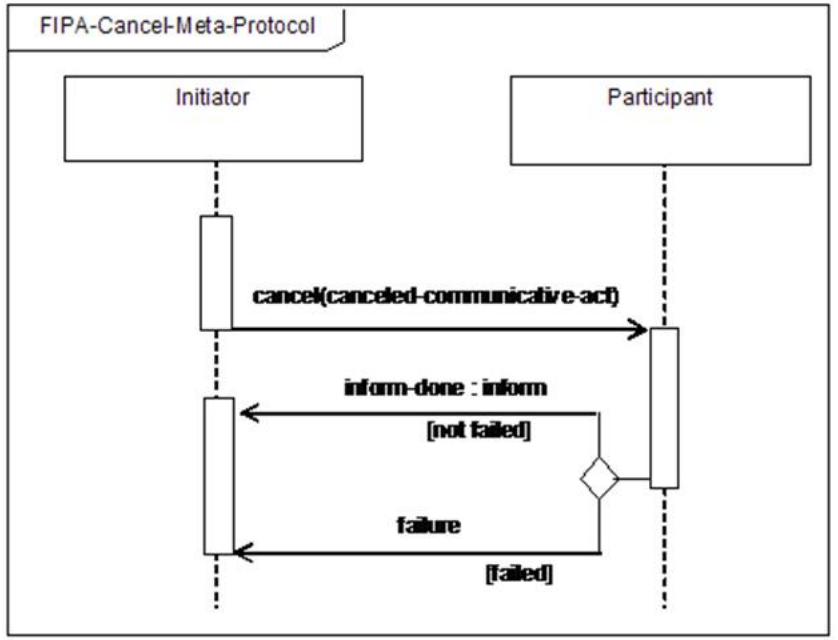
Contract net

Initiator si contractori (potentiali)



FIPA - Contract net





Example

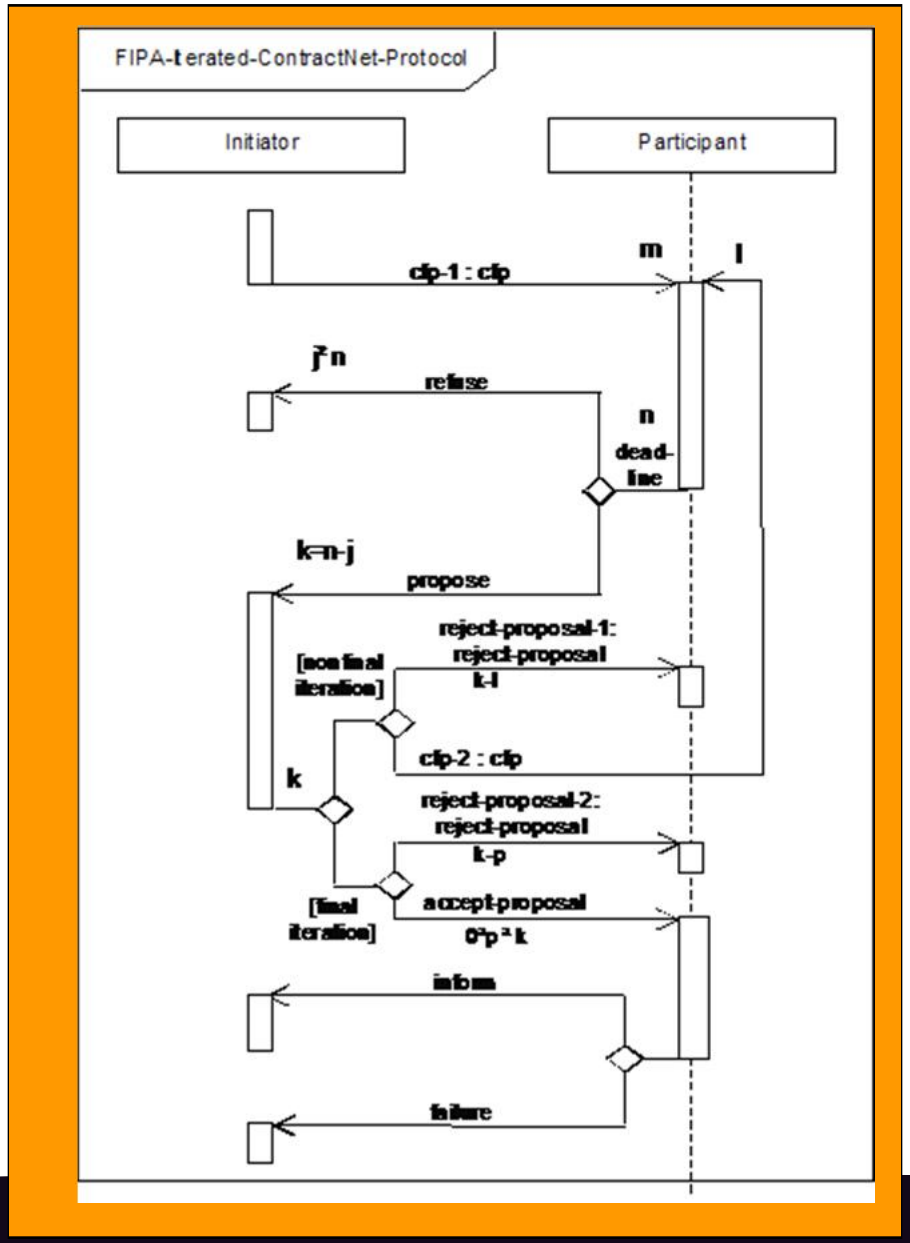
```
(cfp
  :sender (agent-identifier :name j)
  :receiver (set (agent-identifier :name i))
  :content
    "((action (agent-identifier :name i)
      (sell plum 50))
     (any ?x (and (= (price plum) ?x) (< ?x 10))))"
  :ontology fruit-market
  :language fipa-sl)
```

```
(propose
  :sender (agent-identifier :name j)
  :receiver (set (agent-identifier :name i))
  :content
    "((action j (sell plum 50))
     (= (any ?x (and (= (price plum) ?x) (< ?x 10))) 5)"
  :ontology fruit-market
  :in-reply-to proposal2
  :language fipa-sl)
```

```
(accept-proposal
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name j))
  :in-reply-to bid089
  :content
    "   ((action (agent-identifier :name j)
                        (sell plum 50))
                    (= (price plum) 5))) "
  :language fipa-sl)
```

```
(reject-proposal
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name k))
  :content
    "((action (agent-identifier :name k)
              (sell plum 50))
      (= (price plum) 20)
      (price-too-high 20))"
  :in-reply-to bid080)
```

FIPA – Iterated Contract net



Slide 34

a2 adina, 11/22/2007

References

- M. Huhns, L. Stephens. Multiagent systems and societies of agents. In Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, G. Weiss (Ed.), The MIT Press, 2001, p.79-120.
- M. Wooldrige. Reasoning about Rational Agents. The MIT Press, 2000, Chapter 7
- Y. Labrou, T. Finin. Semantics and conversations for an agent communication language. In Readings in Agents, M. Huhns & M. Singh (Eds.), Morgan Kaufmann, 1998, p.235-242.
- J. Ferber - Multi-Agent Systems. Addison-Wesley, 1999, Chapter 6
- T. Finnin, R. Fritzson - KQML as an agent communication language. In Proc. of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press, 1994.
- M. Singh. Agent communication languages: Rethinking the principles. IEEE Computer, Dec. 1998, p.40-47.
- Y. Labrou, T. Finnin, Y. Peng. Agent communication languages: The current Landscape. IEEE Computer, March/April 1999, p. 45-52.
- FIPA97. "Agent Communication Language" Specification FIPA, 11/28/97

Web References

DARPA KSE http://www-ksl.stanford.edu/knowledge-sharing/

KQML http://www.cs.umbc.edu/kqml/

KIF http://logic.stanford.edu/kif/

Ontolingua http://www-ksl-svc.stanford.edu:5915/&service=frame-editor

FIPA http://www.fipa.org/

DAML http://www.daml.org/

References for Ontologies (due to prof. Stefan Trausan)

- Constandache, G.G., Ştefan Trăuşan-Matu, Ontologia şi hermeneutica calculatoarelor, Ed. Tehnică, 2001
- Gruber, T., What is an Ontology, http://www.kr.org/top/definitions.html
- J. Sowa, Ontologia şi reprezentarea cunoştinţelor, în (Constandache şi Trăuşan-Matu, 2001)
- http://www.w3.org/2001/sw/WebOnt/
- http://www.cs.man.ac.uk/~horrocks/Slides/index.html