Lecture #5

Multiagent systems

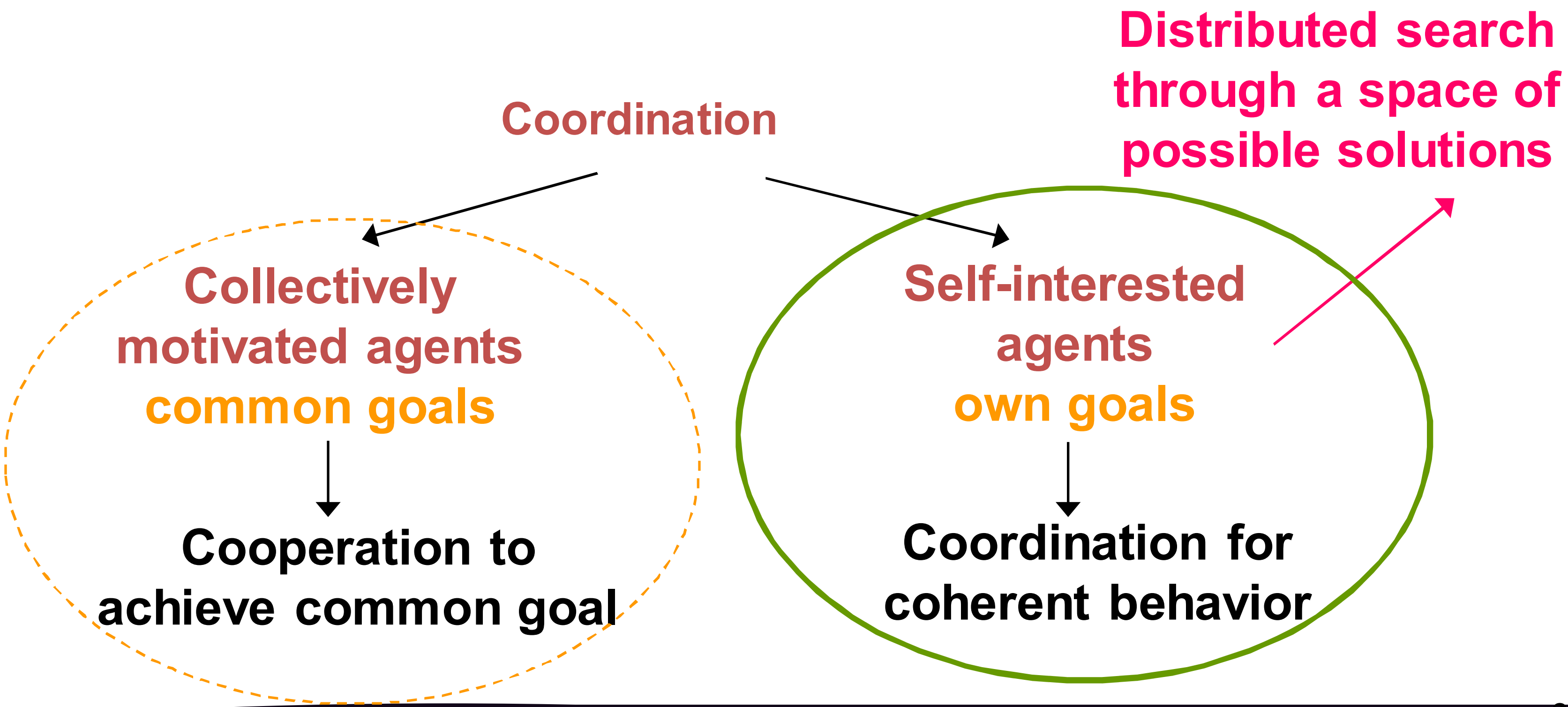
Negotiation techniques in multiagent systems

Lecture outline

- Negotiation principles
- Game theoretic negotiation
  - Evaluation criteria
  - Voting
  - Auctions
- General equilibrium markets
- Task allocation
- Heuristic based negotiation
- Argumentation based negotiation

Negotiation principles

- Negotiation = interaction among agents based on communication for the purpose of coming to an agreement.
- Distributed conflict resolution
- Decision making
- Proposal → accepted, refined, criticized, or refuted



- **Negotiation includes:**
  - a communication language
  - a negotiation protocol
  - a decision process by which an agent decides upon its position, concessions, criteria for agreement, etc.
- Single party **or** multi-party negotiation: one to many or many to many (**eBay** **http://www.ebay.com**)
- May include a single shot message by each party
- **Negotiation techniques**
  - **Game theoretic negotiation**
  - **Heuristic-based negotiation**
  - **Argument-based negotiation**

Game theoretic negotiation

- Utility function
  - $u_i: \Omega \rightarrow R$
  - $\Omega = \{s1, s2, \ldots\}$
  - $u_i(s) \geq u_i(s') \ (s \geq s')$

- Suppose each agent has two possible actions: D and C:
- The environment behaves:

$$t: Ac \times Ac \rightarrow R$$

t(D,D)=r1 t(D,C)=r2 t(C,D)=r3 t(C,C)=r4

or

t(D,D)=r1 t(D,C)=r1 t(C,D)=r1 t(C,C)=r1

u1(r1)=1, u1(r2)=1, u1(r3)=4, u1(r4)=4

u2(r1)=1, u2(r2)=4, u2(r3)=1, u2(r4)=4

u1(D,D)=1, u1(D,C)=1, u1(C,D)=4, u1(C,C)=4

u2(D,D)=1, u2(D,C)=4, u2(C,D)=1, u2(C,C)=4

Agent1 C,C ≥ C,D ≥ D,C ≥ D,D

u1(D,D)=4, u1(D,C)=4, u1(C,D)=1, u1(C,C)=1

u2(D,D)=4, u2(D,C)=1, u2(C,D)=4, u2(C,C)=1

Agent2 D,D $\geq$ D,C $\geq$ C,D $\geq$ C,C

Payoff matrix

		Agent1 Player	
		D	C
Agent2 Player	D	4, 4	4, 1
	C	1, 4	1, 1

Evaluation criteria

- Criteria to evaluate negotiation protocols among self-interested agents
- Agents are supposed to behave rationally
- Rational behavior = an agent prefers a greater utility (payoff) over a smaller one
- Payoff maximization over
  - individual payoffs
  - group payoffs
  - social welfare
- Social welfare
  - The sum of agents' utilities (payoffs) in a given solution.
  - Measures the global satisfaction of the agents
  - Problem: how to compare utilities

- **Pareto efficiency**
  - A solution **x**, i.e., a **payoff vector p(x_1, …, x_n)**, is **Pareto efficient**, i.e., Pareto optimal, if there is no other solution **x'** such that at least one agent is better off in **x'** than in **x** and no agent is worst off in **x'** than in **x**.
  - Measures global satisfaction, does not require utility comparison
  - Social welfare $\subset$ Pareto efficiency
- **Individual rationality (IR)**
  - **IR of an agent participation** = The agent's payoff in the negotiated solution is no less than the payoff that the agent would get by not participating in the negotiation
  - **A mechanism is IR** if the participation is IR for all agents

- **Stability**
  - a protocol is stable if once the agents arrived at a solution they do not deviate from it

Dominant strategy = the agent is best off using a specific strategy no matter what strategies the other agents use

Or:

We say that a strategy $S_1 = \{s_{11}, s_{12}, \ldots, s_{1n}\}$ ***dominates*** another strategy $S_2 = \{s_{21}, s_{22}, \ldots, s_{2m}\}$ if any result of $s \in S_1$ is preferred (best than) to any result of $s' \in S_2$.

Nash equilibrium

- Two strategies, S_1 of agent A and S_2 of agent B are in a **Nash equilibrium** if:
  - in case agent A follows S_1 agent B can not do better than using S_2 **and**
  - in case agent B follows S_2 agent A can not do better than using S_1.
- The definition can be generalized for several agents using strategies S_1, S_2, …, S_k. **The set of strategies** $\{S_1, S_2, …, S_k\}$ used by the agents A_1, A_2, …, A_k is in a **Nash equilibrium** if, for any agent A_i, the strategy S_i is the best strategy to be followed by A_i if the other agents are using strategies $\{ S_1, S_2, …, S_{i-1}, S_{i+1},…, S_k.\}$.

Problems:

- no Nash equilibrum
- multiple Nash equilibria

11

Prisoner's dilema

- Social welfare, Pareto efficient ?
- Nash equilibrium ?

Axelrod's tournament

		Column player	
		Cooperate	Defect
Row player	Cooperate	3, 3	0, 5
	Defect	5, 0	1, 1

Cooperate = not confessing
Defect = confessing

- **Computational efficiency**

To achieve perfect rationality

- The number of options to consider is too big
- Sometimes no algorithm finds the optimal solution

Bounded rationality

- limits the time/computation for options consideration
- prunes the search space

Game of Chicken

		J2 player	
		D	C
J1 player	D	0, 0	3, 1
	C	1, 3	2, 2

Battle of sexes

		Bob player	
		Theater	Football
Anne player	Theatre	2, 1	0, 0
	football	0, 0	1, 2

Coin flip

		J2 jucator	
		Cap	Pajura
J1 jucator	Cap	+1, -1	-1, +1
	Pajura	-1, +1	+1, -1

- We have discussed about pure strategies
- A **mixed strategy** is an assignment of a probability to each pure strategy
- A **mixed strategy** $\boldsymbol{p_i}$ of a player $\boldsymbol{i}$ is a probability distribution over actions $\boldsymbol{A_i}$ available to $\boldsymbol{I}$
- A **pure Nash equilibrium** is a Nash equilibrium using pure strategies
- A **mixed Nash equilibrium** is a Nash equilibrium using mixed strategies
- A mixed Nash equilibrium is a set of mixed strategies, one for each player, so that no player has an incentive to unilaterally deviate from their assigned strategies

14

		J2 player	
		L	R
J1 player	T	2, 1	0, 0
	B	0, 0	1, 2

		J2 player	
		L	R
J1 player	T	p * q	p * (1-q)
	B	(1-p) * q	(1-p) * (1-q)

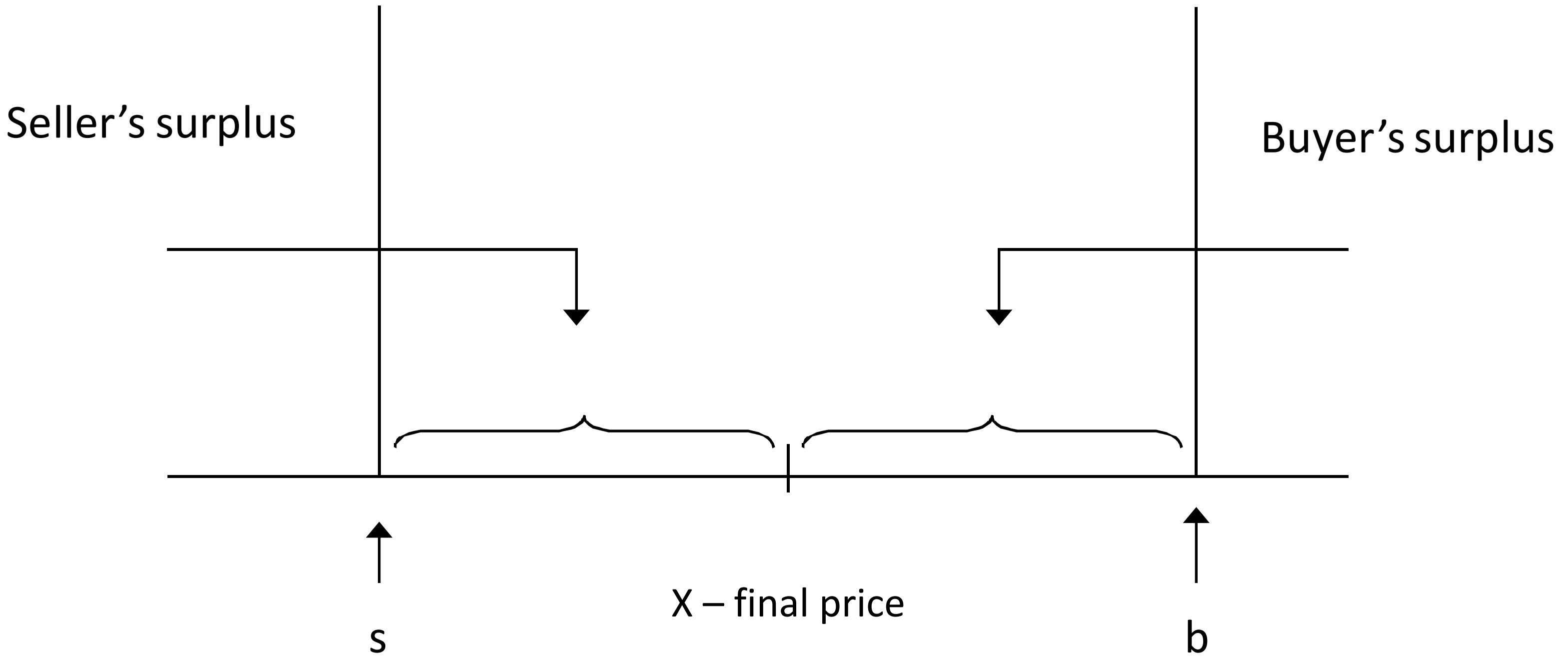
15

Bargain

- In a transaction when the seller and the buyer value a product differently, a surplus is created. A bargaining solution is then a way in which buyers and sellers agree to divide the surplus.
- A – house 10, B – house 20
- Trade would result in the generation of surplus, whereas no surplus is created in case of no-trade.
- Bargaining Solution provides an acceptable way to divide the surplus among the two parties.

16

Bragain



Seller’s RP
Seller wants s or more

Buyer’s RP
Buyer wants b or less

17

- **A Bargaining Solution is defined:**
- $F : (X,d) \rightarrow S$,
  - $X \subseteq R^2$ and $S,d \subseteq R^2$.
  - X represents the utilities of the players in the set of possible bargaining agreements.
  - d represents the **point of disagreement**.

- price $\in$ [10,20], bargaining set is simply $x + y \leq 10$, $x \geq$, $y \geq 0$.
- A point (x,y) in the bargaining set represents the case, when seller gets a surplus of x, and buyer gets a surplus of y, i.e. seller sells the house at 10 + x and the buyer pays 20 - y.

The Ultimatum Game

- **P1** proposes how to divide the sum **x** between the two players: **p** and **p-x**
- **P2** can either **accept** or **reject** this proposal (f(p) = *accept* or *reject*)
- If P2 *accepts*, the money is split according to the proposal.
  - **P1** gets **p** and **P2** gets **p-x**
- If **P2** rejects, neither player receives anything.

19

The Ultimatum Game

- (p, f) is a **Nash equilibrium** of the Ultimatum Game if
- $f(p)$ = "accept" and there is no $y > p$ such that $f(y)$ = "accept" (i.e. player 2 would reject all proposals in which player 1 receives more than p).
- The first player would not want to unilaterally increase his demand since the second will reject any higher demand.
- The second would not want to reject the demand, since he would then get nothing.

20

Voting

Truthful voters

- Rank feasible social outcomes based on agents' individual ranking of those outcomes
- A - set of *n* agents
- O - set of *m* feasible outcomes
- Each agent has a preference relation $<_i$: O x O, asymmetric and transitive

Social choice rule

- *Input:* the agents' preference relations ($<_1$, …, $<_n$)
- *Output:* elements of O sorted according the input - gives the social preference relation <*

21

Properties of the social choice rule:

- A social preference ordering $<^*$ should exist for all possible inputs (individual preferences)
- $<^*$ should be defined for every pair $(o, o') \in O$
- $<^*$ should be asymmetric and transitive over O
- The outcomes should be Pareto efficient:

$$\text{if } \forall i \in A,\ o <_i o' \text{ then } o <^* o'$$

- No agent should be a dictator in the sense that

  $o <_i o'$ implies $o <^* o'$ for all preferences of the other agents

Arrow's impossibility theorem

- No social choice rule satisfies all of the six conditions

Binary protocol

Pluralist protocols

23

Binary protocols

- 35% agents c>d>b>a
- 33% agents a>c>d>b
- 32% agents b>a>c>d

- Agenda 1: (b,d), d, (d,a) a, (c,a) a
- Agenda 2: (c,a) a, (d,a) a, (a,b) b
- Agenda 3: (a,b) b, (b,c) c (c,d) c
- Agenda 4: (c,a) a (a,b) b, (b,d) d

24

Pluralist protocols

Borda protocol = assigns an alternative |O| points for the highest preference, |O|-1 points for the second, and so on

- The counts are summed across the voters and the alternative with the highest count becomes the social choice

25

Borda Protocol

Agent	Preference
1	a>b>c>d
2	b>c>d>a
3	c>d>a>b
4	a>b>c>d
5	b>c>d>a
6	c>d>a>b
7	a>b>c>d

Agent	Preference
1	a>b>c
2	b>c>a
3	c>a>b
4	a>b>c
5	b>c>a
6	c>a>b
7	a>b>c

- c gets 20, b 19, a 18, d 13
- elim d – a 15, b 14, c 13

Winner turns loser and loser turns winner if the lowest ranked alternative is removed

26

Auctions

(a) Auction theory = agents' protocols and strategies in auctions

- The auctioneer wants to sell an item at the highest possible payment and the bidders want to acquire the item at the lowest possible price
- A centralized protocol, includes one auctioneer and multiple bidders
- The auctioneer announces a good for sale. In some cases, the good may be a combination of other goods, or a good with multiple attributes
- The bidders make offers. This may be repeated for several times, depending on the auction type
- The auctioneer determines the winner

27

- Auction characteristics:
  - Simple protocols
  - Centralized
  - Allows collusion “behind the scenes”
  - May favor the auctioneer

(b) Auction settings

- **Private value auctions**: the value of a good to a bidder agent depends only on its private preferences. Assumed to be known exactly
- **Common value auctions**: the good’s value depends entirely on other agents’ valuation
- **Correlated value auctions**: the good’s value depends on internal and external valuations

(c) Auction protocols

English (first-price open cry) auction - each bidder announces openly its bid; when no bidder is willing to raise anymore, the auction ends. The highest bidder wins the item at the price of its bid.

Strategy:

- In private value auctions the dominant strategy is to always bid a small amount more than the current highest bid and stop when the private value is reached.
- In correlated value auctions the bidder increases the price at a constant rate or at a rate it thinks appropriate

First-price sealed-bid auction - each bidder submits one bid without knowing the other's bids. The highest bidder wins the item and pays the amount of his bid.

Strategy:

- No dominant strategy
- Bid less than its true valuation but it is dependent on other agents bids which are not known

Dutch (descending) auction - the auctioneer continuously lowers the price until one of the bidders takes the item at the current price.

Strategy:

- Strategically equivalent to the first-price sealed-bid auction
- Efficient for real time

Vickery (second-price sealed-bid) auction - each bidder submits one bid without knowing the other's bids. The highest bid wins but at the price of the second highest bid

Strategy:

- The bidder dominant strategy is to bid its true valuation

All-pay auctions - each participating bidder has to pay the amount of his bid (or some other amount) to the auctioneer

30

(d) Problems with auction protocols

- They are not collusion proof
- Lying auctioneer
  - Problem in the Vickery auction
  - Problem in the English auction - use shills that bid in the auction to increase bidders' valuation of the item
  - The auctioneer bids the highest second price to obtain its reservation price – may lead to the auctioneer keeping the item
  - Common value auctions suffers from the winner's curse: agents should bid less than their valuation prices (as winning the auction means its valuation was too high)
  - Interrelated auctions – the bidder may lie about the value of an item to get a combination of items at its valuation price

3. General equilibrium market mechanisms

- General equilibrium theory =

  a microeconomic theory
- *n* commodity goods *g*, $g = 1,n$, amount unrestricted
- prices **$p=[p_1, \dots, p_n]$,** where $p_g \in R$ is the price of good *g*
- 2 types of agents: **consumers** and **producers**

32

- 2 types of agents: consumers and producers

Consumers:

- An **utility function $u_i(x_i)$** which encodes its preferences over different consumption bundles **$x_i=[x_{i1},\ldots,x_{in}]$,** where $x_{ig} \in R^+$ is the consumer's *i*'s allocation of good *g*.
- An initial endowment **$e_i=[e_{i1},\ldots,e_{in}]$,** where e_{ig} is its endowment of commodity *g*

Producers:

- Production vector **$y_j=[y_{j1},\ldots,y_{jn}]$** where y_{jg} is the amount of good *g* that producer *j* produces
- Production possibility set **Y_j** - the set of feasible production vectors

33

- The **profit** of producer *j* is $\mathbf{p} \cdot \mathbf{y_j}$, where $y_j \in Y_j$.
- The producer's profits are divided among the consumers according to predetermined proportions which need not be equal.
- Let θ_{ij} be the fraction of producer *j* that consumer *i* owns
- The producers' profits are divided among consumers according to these shares
- Prices may change and the agents may change their consumption and production plans but

  - actual production and consumption only occur when **the market has reached a general equilibrium**

($\mathbf{p}^*$, $\mathbf{x}^*$, $\mathbf{y}^*$) is a **Walrasian equilibrium** if:

- **markets clear**

$$\sum_i x^*{}_i = \sum_i e_i + \sum_j y^*{}_j$$

- **each consumer *i* maximizes its preferences given the prices**

$$x^*{}_i = \arg \max_{x_i \in R_n{}^+,\, p^*.x_i \le p^*.e_i + \sum_j \theta_{ij} p^*.y_j} u_i(x_i)$$

- **each producer *j* maximizes its profits given the prices**

$$y^*{}_j = \arg\max_{y_j \in Y_j} p^*.y_j$$

Properties of Walrasian equilibrium:

- **Pareto efficiency** - the general equilibrium is Pareto efficient, i.e., no agent can be made better off without making some other agent worse off
- **Coalitional stability** - each general equilibrium is stable in the sense that no subgroup of consumers can increase their utilities by pulling out the equilibrium and forming their own market
- **Uniqueness under gross substitutes** - a general equilibrium is unique if the society-wide demand for each good is nondecreasing

36

The distributed price tatonnement algorithm

This image cannot currently be displayed.

Algorithm for price adjustor:

1. $p_g=1$ for all $g \in [1..n]$
2. Set λ_g to a positive number for all $g \in [1..n]$
3. **repeat**

    3.1 Broadcast **p** to consumers and producers

    3.2 Receive a production plan y_j from each producer *j*

    3.3 Broadcast the plans $\mathbf{y_j}$ to consumers

    3.4 Receive a consumption plan $\mathbf{x_i}$ from each consumer *i*

    3.5 **for** g=1 to n **do**

$$p_g = p_g + \lambda_g(\Sigma_i(x_{ig} - e_{ig}) - \Sigma_j y_{jg})$$

until $|\Sigma_i(x_{ig}-e_{ig})- \Sigma_j y_{jg}| < \varepsilon$ for all $g \in [1..n]$

4. Inform consumers and producers that an equilibrium has been reached

The distributed price tatonnement algorithm

This image cannot currently be displayed.

Algorithm for consumer *i*:

1. repeat

1.1 Receive **p** from the adjustor

1.2 Receive a production plan $\mathbf{y_j}$ for each *j* from the adjustor

1.3 Announce to the adjustor a consumtion plan $\mathbf{x_i} \in R^n_+$ that maximizes $u_i(x_i)$ given the budget constraint

$$\mathbf{p.x_i} \leq \mathbf{p.e_i} + \sum_j \theta_{ij} \mathbf{p.y_j}$$

until informed that an equilibrium has been reached

2. Exchange and consume

Algorithm for producer *j*:

1. repeat

1.1 Receive p from the adjustor

1.2 Announce to the adjustor a production plan $\mathbf{y_j} \in Y_j$ that maximizes $\mathbf{p.y_j}$

until informed that an equilibrium has been reached

2. Exchange and produce

38

4. Task allocation through negotiation

This image cannot currently be displayed.

General equilibrium market mechanisms use

- global prices
- a centralized mediator

Drawbacks:

- not all prices are global
- bottleneck of the mediator
- mediator - point of failure
- agents have no direct control over the agents to which they send information

Need of a more distributed solution

39

4.1 Task allocation by redistribution

This image cannot currently be displayed.

- A **task-oriented domain** is a triple **<T, Ag, c>** where
  - **T** is a set of tasks;
  - **Ag** = {1, . . . ,n} is a set of agents which participate in the negotiation;
  - **c:$\mathcal{P}$(T) $\rightarrow$ R$^+$** is a cost function which defines the costs for executing every sub-set of tasks
- The cost function must satisfy two constraints:
  - must be monotone
  - the cost of a task must not be 0, i.e., c(Φ) = 0.
- An **encounter** within a task-oriented domain

  <T, Ag, c> occurs when the agents *Ag* are assigned tasks to perform from the set T
- It is an assignment of tasks R = {E_1, . . ., E_n}, $E_i \subseteq$ T,

  i $\in$Ag, to agents Ag

- Encounter: can an agent be better off by a task redistribution? **Deal**

  This image cannot currently be displayed.

  Example:

  Ag = $\{a_1, a_2, a_3\})$ T = $\{t_1, t_2, t_3, t_4, t_5\}$

  Encounter

  R = $\{E_1, E_2, E_3\}$ with $E_1 = \{t_1, t_3\}$, $E_2 = \{t_2\}$, $E_3 = \{t_4, t_5\}$

  Deal

  α = $\{D_1, D_2, D_3\}$ with $D_1 = \{t_1, t_2\}$, $E_2 = \{t_3, t_4\}$, $E_3 = \{t_5\}$

- The cost of a deal α for agent a_1 is $c(D_1)$ and the cost for a_2 is $c(D_2)$.
- **The utility of a deal** represents how much the agents should gain from that deal

  $utility_i(\alpha) = c_i(E_i) - c_i(D_i)$, for i = 1, 2, 3

- A deal α_1 is said to **dominate** another deal α_2 if and only if:
  - Deal α_1 is at least as good for every agents as α_2

    $\forall\ i \in \{1,2\}\ \text{utility}_i(\alpha_1) \geq \text{utility}_i(\alpha_2)$
  - Deal α_1 is better for some agent than α_2

    $\exists\ i \in \{1,2\}\ \text{utility}_i(\alpha_1) > \text{utility}_i(\alpha_2)$

This image cannot currently be displayed.

- A deal **weakly domintaes** another deal if (1) is fulfilled
- If a deal is not dominated by any other deal then the deal is **Pareto optimal**
- **Task re-allocation = finding a Pareto optimal deal**

- Task allocation improves at each step ~ hill climbing in the space of task allocations where the height-metric of the hill is social welfare
- It is an anytime algorithm
  - Contracting can be terminated at anytime
  - The worth of each agent's solution increases monotonically $\rightarrow$ social welfare increases monotonically

42

Monotonic concession protocol

Several negotiation rounds (u)

1. u $\leftarrow$ 1, a1 and a2 propose deals from the negotiation set: α_1 and α_2
2. **if** a1 proposes α_1 **and** a2 proposes α_2 such that:
   (i) **utility$_1$(α_2) $\geq$ utility$_1$(α_1)**
   or
   (ii) **utility$_2$(α_1) $\geq$ utility$_2$(α_2)**
   then agreement is reached **stop**
3. **else** u $\leftarrow$ u+1
4. **if** a1 proposes α_1 **and** a2 proposes α_2 such that:
   $\text{utility}_1(\alpha_2^{u}) \geq \text{utility}_1(\alpha_2^{u-1})$ **and**
   $\text{utility}_2(\alpha_1^{u}) \geq \text{utility}_1(\alpha_1^{u-1})$
   then go to 2
5. **else** negotiation ends in conflict **stop**

- *IR contract*
- *Problem*: task allocation stuck in a local optimum = no contract is individually rational (IR) and the task allocation is not globally optimal
- *Possible solution*: different contract types:
  - O – one task
  - C – cluster contracts
  - S – swap contracts
  - M – multi-agent contracts

- For each 4 contract types (O, C, S, M) there exists task allocations for which there is an IR contract under one type but no IR contracts under the other 3 types
- Under all 4 contract types there are initial task allocations for which no IR sequence of contracts will lead to the optimal solution (social welfare)

Main differences as compared to game theoretic negotiation

- An agent may reject an IR contract
- An agent may accept a non-IR contract
- The order of accepting IR contracts may lead to different pay offs
- Each contract is made by evaluating just a single contract instead of doing lookahead in the future

Un-truthful agents

- An agent may lie about what tasks it has:
  - Hide tasks
  - Phantom tasks
  - Decoy tasks
- Sometimes lying may be beneficial

45

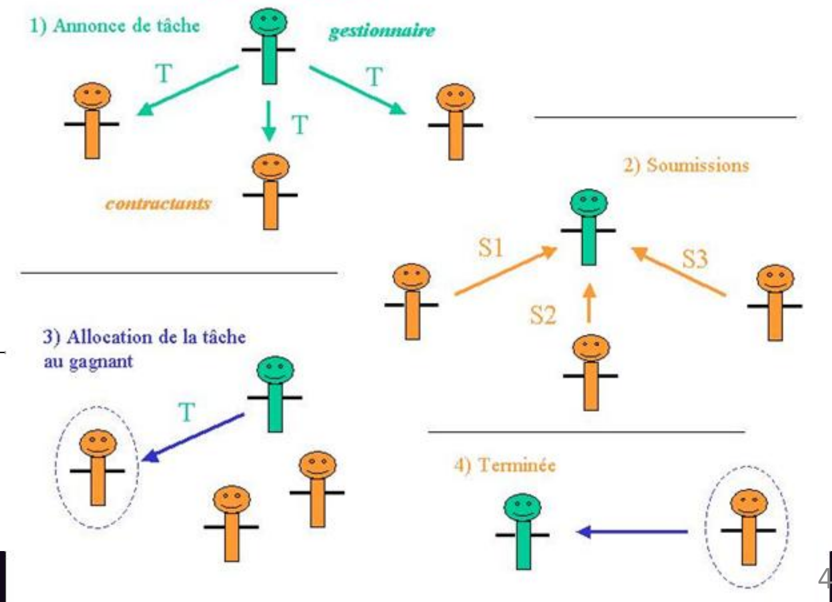
4.2 Contract Net

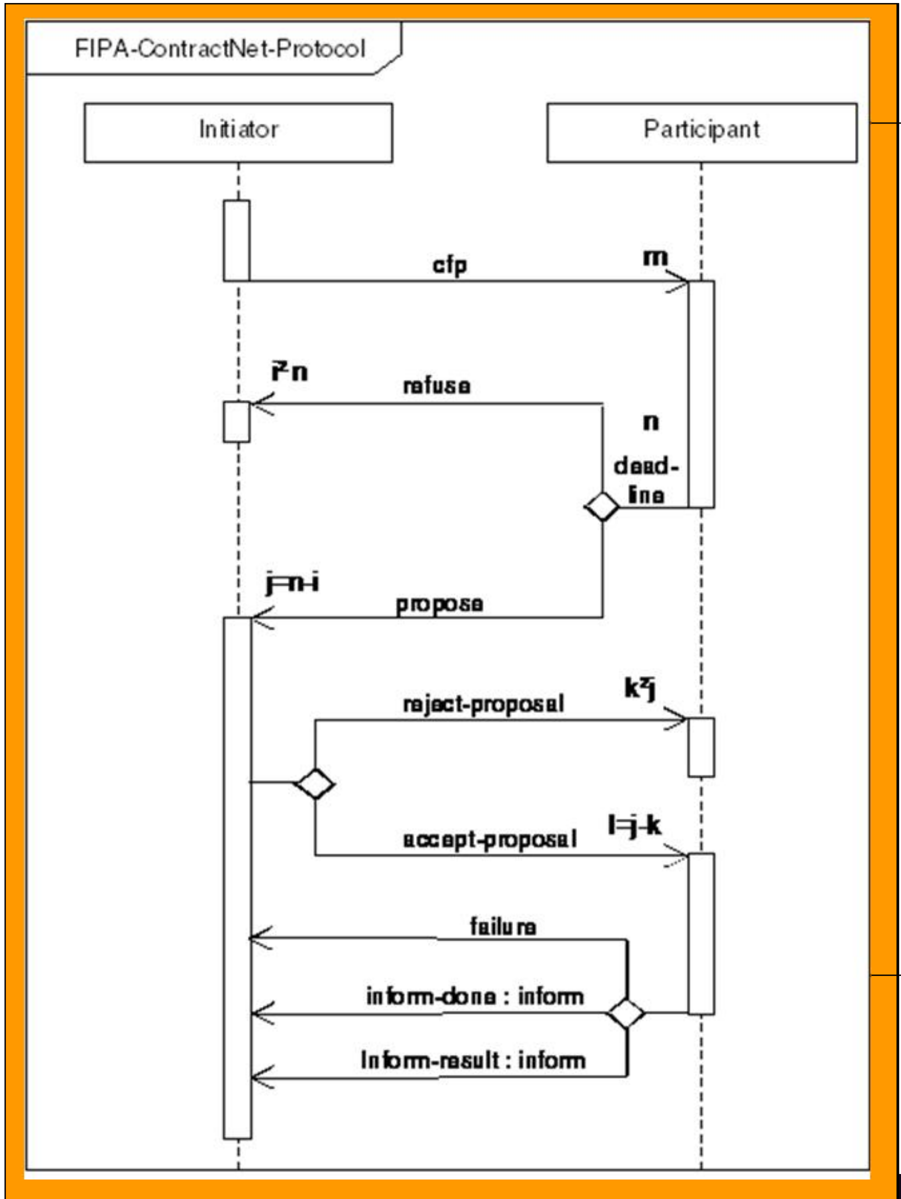
This image cannot currently be displayed.

Task allocation via negotiation - Contract Net

- A kind of bridge between game theoretic negotiation and heuristic-based one
- In a Contract Net protocol, the agents can have two roles: **contractor** (initiator) or **bidder** (participant)

Protocol implemented in FIPA





This protocol is identified by the token **fipa-contract-net** as the value of the **protocol** parameter of the ACL message.

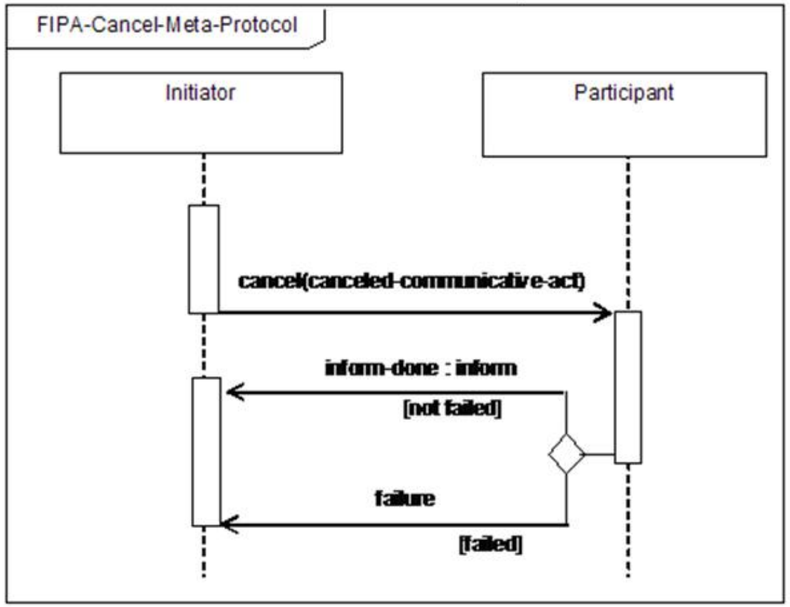


Diagram - extensions to UML1.x. [Odell2001]

Agent *j* asks agent *j* proposals for selling 50 plum boxes and price conditions

```
(cfp
  :sender (agent-identifier :name j)
  :receiver (set (agent-identifier :name i))
  :content
   "((action (agent-identifier :name i)
        (sell plumbox 50))
        (any ?x (and (= (price plumbox) ?x) (< ?x 10))))"
  :ontology fruit-market
  :language fipa-sl
  :protocol fipa-contract-net
  :conversation-id c007
  :reply-by 10)
```

Agent *j* answers to i

(**propose**

:**sender** (agent-identifier :name j)

:**receiver** (set (agent-identifier :name i))

:**in-reply-to** proposal2

:**content**

"((action j (sell plumbox 50))

(= (any ?x (and (= (price plumbox) ?x) (< ?x 10))) 5)"

:**ontology** fruit-market

:**language** fipa-sl

:**protocol** fipa-contract-net

:**conversation-id** c007)

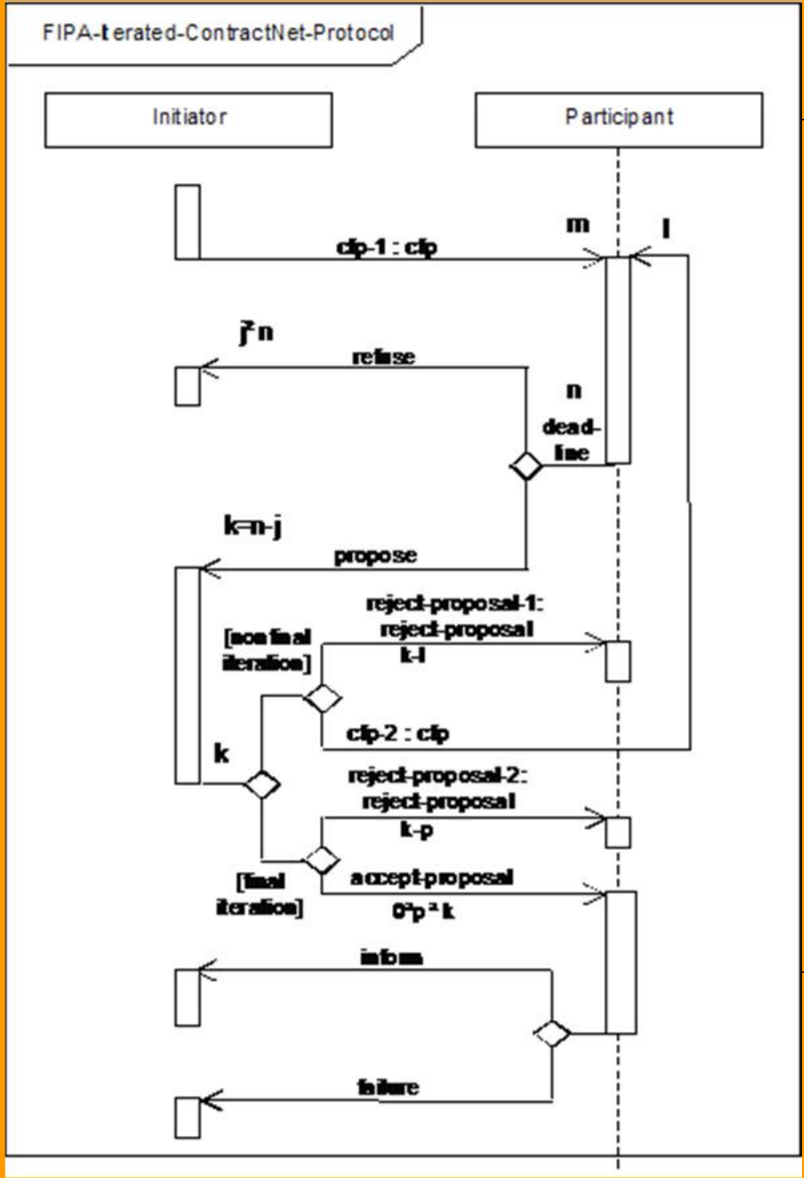
Agent *i* accepts proposal of j

```
(accept-proposal
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name j))
  :in-reply-to bid089
  :content
   "   ((action (agent-identifier :name j)
                  (sell plumbox 50))
              (= (price plumbox) 5))) "
  :ontology fruit-market
  :language fipa-sl
  :protocol fipa-contract-net
  :conversation-id c007)
```

Agent *i* refuses the proposal of k

```
(reject-proposal
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name k))
  :in-reply-to bid080
  :content
   "((action (agent-identifier :name k)
          (sell plumbox 50))
          (= (price plumbox) 20)
          (price-too-high 20))"
  :ontology fruit-market
  :language fipa-sl
  :protocol fipa-contract-net
  :conversation-id c007)
```

FIPA – Iterated Contract net



This protocol is identified by the token **fipa-iterated-contract-net** as the value of the **protocol** parameter of the ACL message.

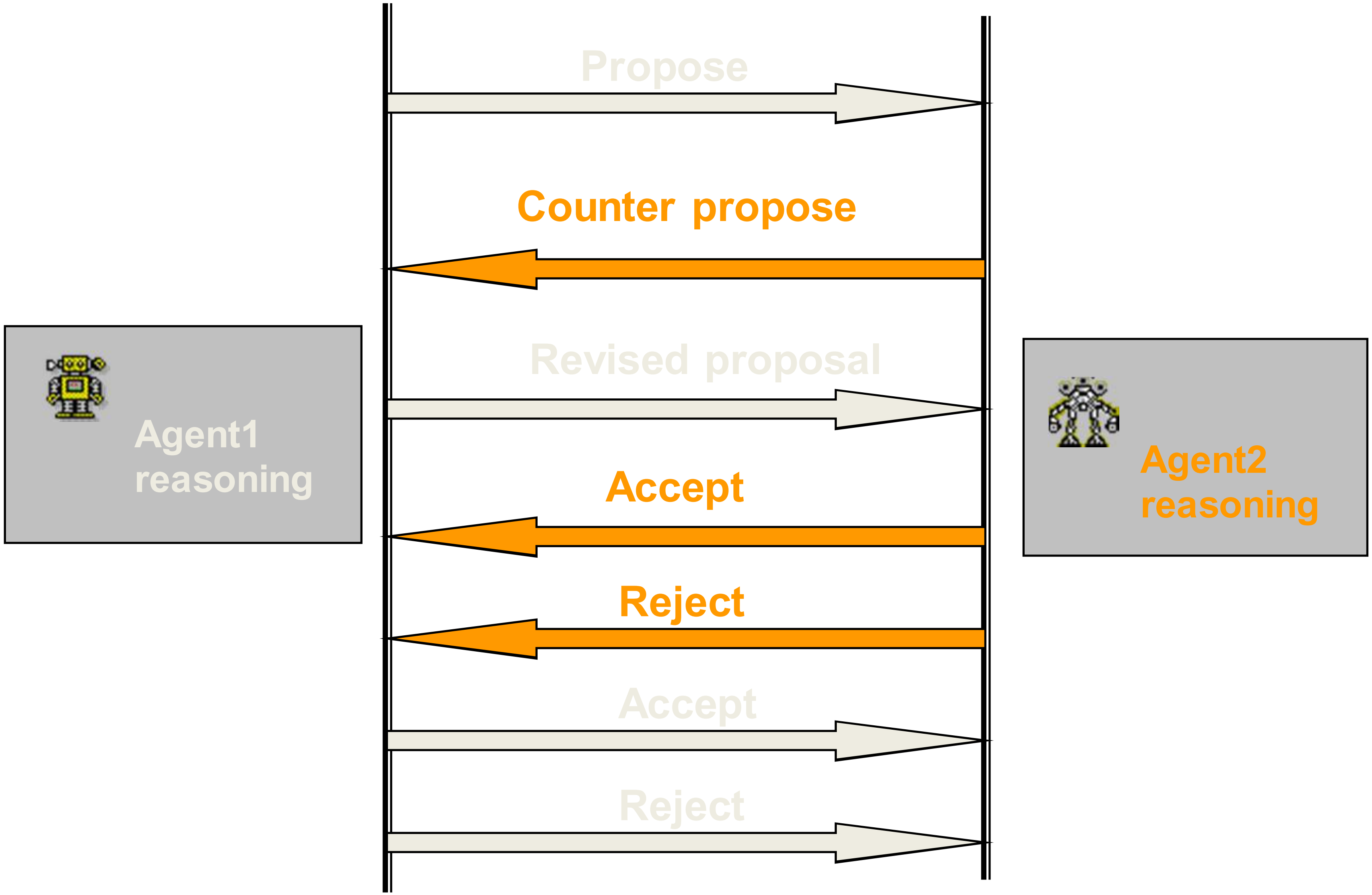
Slide 52

a3 adina, 11/22/2007

5 Heuristic-based negotiation

- Produce a good rather than optimal solution
- Heuristic-based negotiation:
  - Computational approximations of game theoretic techniques
  - Informal negotiation models
- No central mediator
- Utterances are private between negotiating agents
- The protocol does not prescribe an optimal course of action
- Central concern: the agent’s decision making heuristically during the course of negotiation

53


Propose
Counter propose
Revised proposal
Accept
Reject
Accept
Reject
Agent1 reasoning
Agent2 reasoning

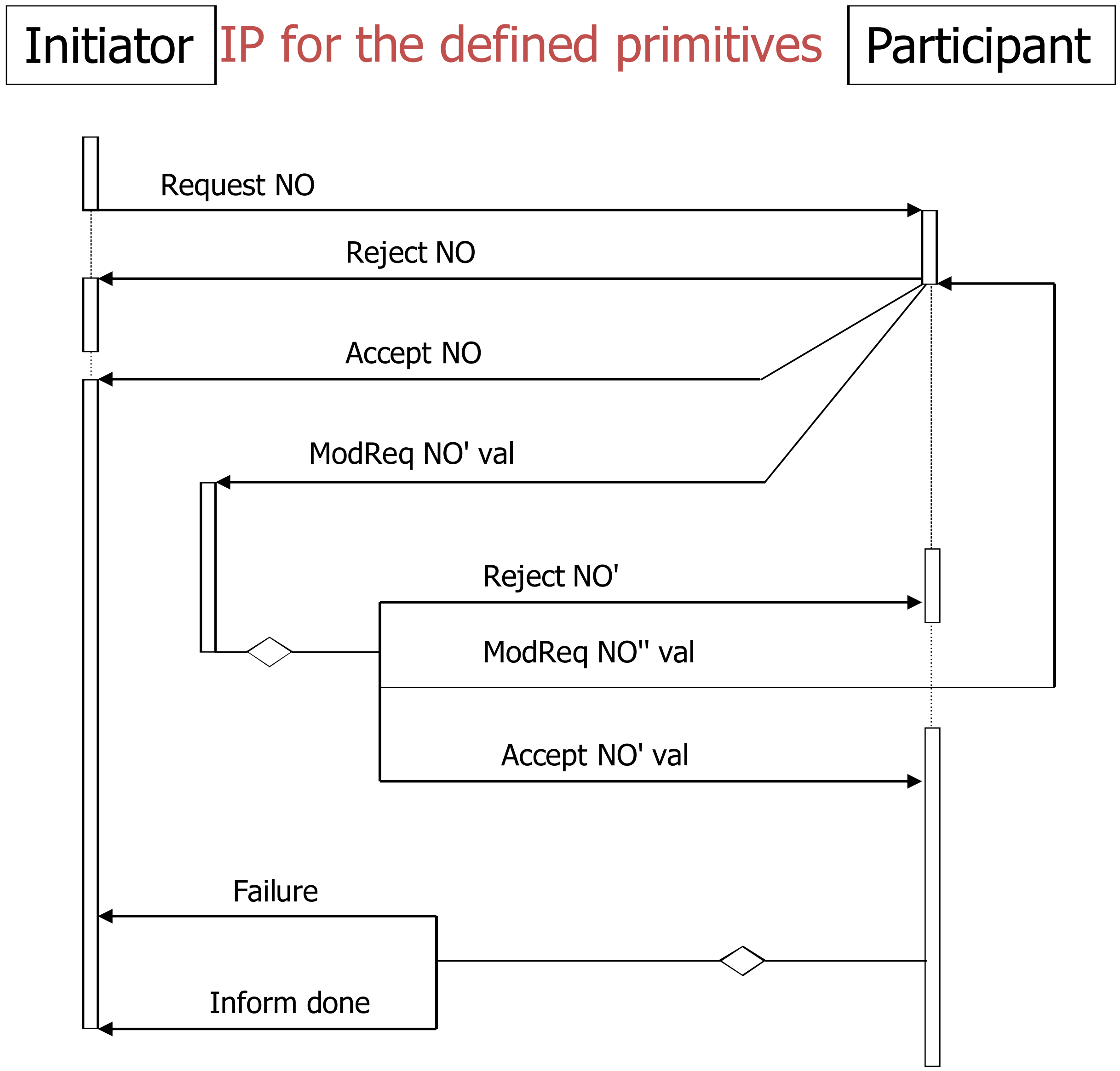
- A ***negotiation object*** **(NO)** is the range of issues over which agreements must be reached
- The object of a negotiation may be an action which the negotiator agent *A* asks another agent *B* to perform for it, a service that agent *A* asks to *B*, or, alternately, an offer of a service agent *A* is willing to perform for *B* provided *B* agrees to the conditions of *A*.

NO03: NO

- **Name:** Paint_House
- **Cost:** Value:100, Type: integer, Modif=Yes;
- **Deadline:** Value: May_12, Type: date, Modif=No;
- **Quality:** Value: high, Type: one of (low, average, high), Modif=Yes

- ☐ **(Request NO)** - request of a negotiation object
- ☐ **(Accept name(NO))** - accept the request for the NO
- ☐ **(Reject name(NO))** - reject the request for the NO
- ☐ **(ModReq name(NO) value(NO,X,V1))** - modify the request by modifying the value of the attribute X of the NO to a different value V1

55

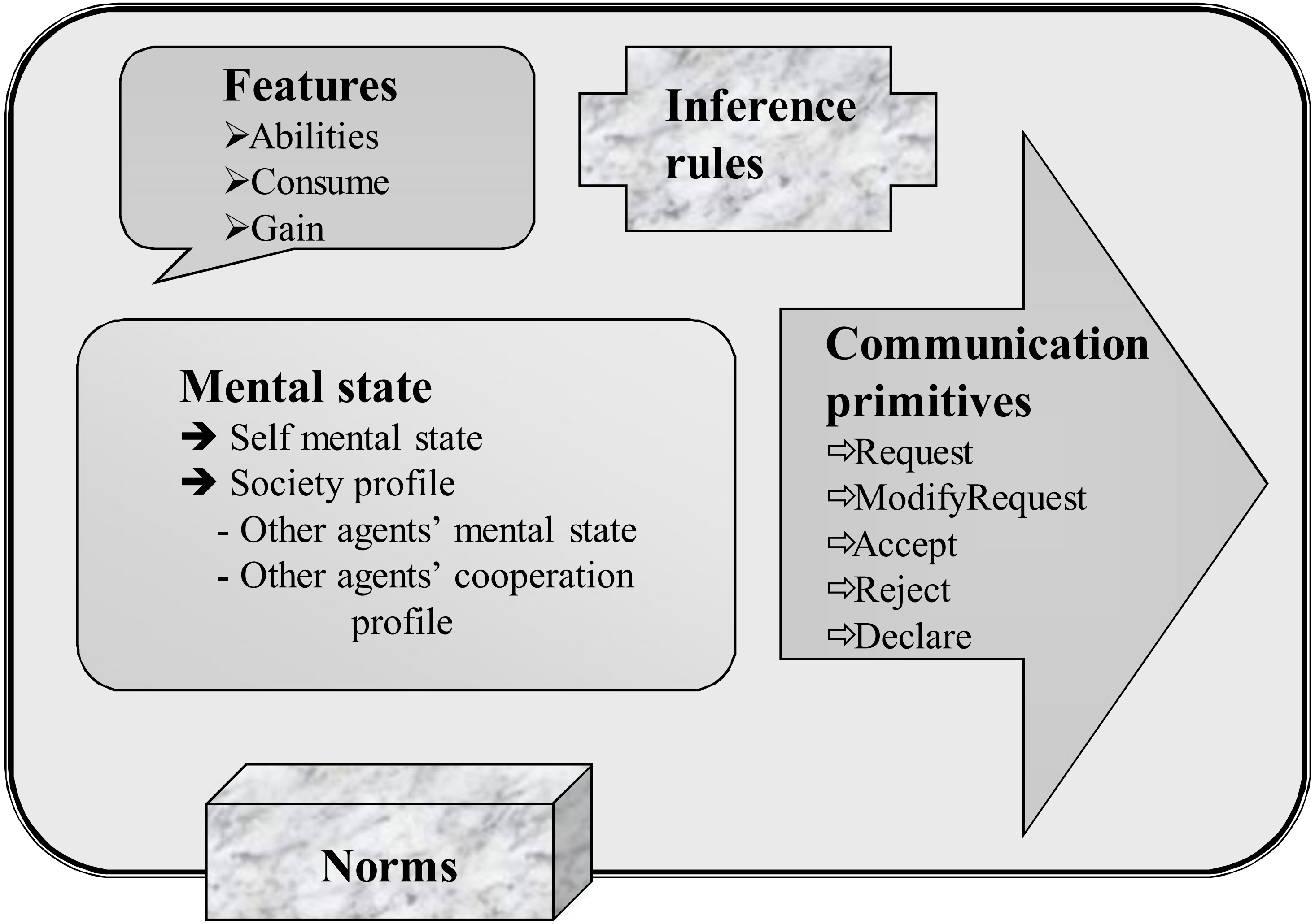

Initiator
IP for the defined primitives
Participant
Request NO
Reject NO
Accept NO
ModReq NO' val
Reject NO'
ModReq NO'' val
Accept NO' val
Failure
Inform done

56

Example

- Model of a MAS with self-interested agents aiming to:
- achieve their own goals
- comply to obligations and norms
- obtain maximum gain
- establish good cooperation relationships in the society

Agent Representation



The Mental Model
Mental state - self

- Beliefs - $Bel_i w$
- Desires - $Des_i w$
- Intentions - $Int_i w$

BDI model

- $Goals_i \subseteq \{Des_i w\}$

Intentions-to (agent)
Intensions-that (others)

- Obligations - $Ob_i w$
- Preferences - $Pref_i(w,v)$ - i prefers w with value v

Agent Features

- Abilities - $Ab_i w$
- Consumes - $Cons_i(w,v)$ - agent i consumes v for executing the action w
- Gain - $Gain_i(w,v)$ - agent i gains v for achieving goal w

Norms

- permitted actions in MAS

Communication Primitives

- Request(*w*, DeadLine, Payment)
- ModifyRequest(*w*, DeadLine, Payment)
- Accept(*w*, DeadLine, Payment)
- Reject(*w*, Justification)
- Declare(*w*)

Messages

- Send: $Ag \times Ag \longrightarrow M$
- Receive: $Ag \times Ag \longrightarrow M$

Agent Reasoning

- Reasoning capabilities

about the world state

how to select goals

how to achieve goals

how to conduct negotiation

based on → gain

→ cooperation profile

Inference Rules

- Inference Rules for updating the Mental State
- Inference Rules for goal Selection
- Inference Rules for plan generation
- Inference Rules for evaluating the cooperation profile
- Inference Rules for negotiation
  - (a) Request generation & selection
  - (b) Incoming request evaluation & answer generation
  - (c) Answer evaluation & reply generation

Agent Control Structure

- 2 Phases

→ **Phase I**: Control of agent's activities which do not depend on other agents

→ **Phase II**: Negotiation and reaching of agreements

Phase I

- Select Goals as a non-contradictory subsets of Desires
- Generate Plans for achieving Goals
- Analyze Plans from the point of view of norm compliance
- <u>if</u> actions in Plans violate Norms
  <u>then</u> revise Plans or revise Goals
- <u>if</u> there are intentions-that
- <u>then</u> search descriptions of other agents <u>and</u> identify the agents {i} with $Ab_{\{i\}}$ able to do ***intentions-that***
  - <u>if</u> no such agents exist
  - <u>then</u> address Facilitator or revise Plans or revise Goals
- Perform all ***intentions-to***

Phase II

- Generate requests for agents in {i} to do ***intentions-that***
- Select requests $\{Req_{\{i\}}\}$ to be sent
- Send requests $\{Req_{\{i\}}\}$
- Read answers to $\{Req_{\{i\}}\}$
- Evaluate answers, accept them or generate counterproposals
- Evaluate incoming requests $\{Req_A\}$ and generate answers
- Update mental model
- Send answers to $\{Req_A\}$ (accept or counterproposals)

Cooperation profile of agent x as seen by A

- No. of A’s requests accepted by *x* (No_req)
- No. of *A*’s requests rejected by *x* (No_reject)
- *A*’s gain obtained from *x*’s previous actions (My_gain)
- *x*’s credit as given by *A* (Given_credit)
- *A*’s credit as given by *x* (My_credit)
- No. of *x*’s abilities that may lead to *A*’s goal fulfillment (No_abil)

Negotiation

(a) Request generation & selection rules

- Generate
  (ListOfAgents (Action=N DeadLine Payment))
- Apply rules to compute Payment and rank the agents, based on the gain for executing Action *N* and on the cooperation profile

g_N - the gain of *N* computed from $Gain_A(w, v)$
Pmax - maximum gain for action *N*

if Action = *N* **and** Max Payment.*N* = Pmax
and *x* **isin** ListOfAgents **and** No_req.*x* > 0
and My_gain.*x* > 0 **and** Given_credit.*x* > 0
then Rank.*x* = 4 **and** Payment.*N* = Pmax/2

- Choose agent/agents **x** with the highest Rank.x

>> Send(*A*, *x*) = Request(*N*, DeadLine, Payment)

(b) Incoming request evaluation & answer generation rules

Request received

Receive(A, x) = Request(N, DeadLine, Payment)

- Check $Ab_A N$ for action N
- Check compliance of N to Norms

>> Send(A, x) = Reject(N, Justification)

Justification → NotAbility
Justification → NotConfNorms

- Check consistency of N with ObA and GoalsA
- Payment > $Cons_A$(N, Cost) ?
- Check possibility to meet DeadLine

>> Send(A, x) = Accept(N, DeadLine, Payment)

- A adopts N as one of its current intentions

(b) Incoming request evaluation & answer generation rules

Payment < $Cons_A(N, \text{Cost})$?

if Action = *N* **and** Consume.*N* = Cost
and Cost > Payment **and** No_req.*x* > 0
and My_gain.*x* > 0 **and** My_credit.*x* > 0
then Rank.*x* = 4 **and** Given_credit.*x* = Cost - Payment

- Rank the agent
- if the rank is above a certain value
  then update the cooperation profile

Given_credit.x = Cost - payment

>> Send(*A*, *x*) = Accept(*N*, DeadLine, Payment)

or

>> Send(*A*, *x*) = ModifyRequest(*N*, DeadLine, Payment1)

(c) Answer evaluation & reply generation rules

- *Acceptance answer received*

<u>Receive(*A*, *x*) = Accept(*N*, DeadLine, Payment)</u>

- End negotiation and update cooperation profile

- *Rejection answer received*

<u>Receive(*A*, *x*) = Reject(*N*, Justification)</u>

- End negotiation, update cooperation profile and mental state

- *Counterproposal answer received*

<u>Receive(*A*, *x*) = ModifyRequest(*N*, DeadLine1, Payment1)</u>

- Use (b) set of rules

6 Argumentation-based negotiation

- **Arguments** used to persuade the party to accept a negotiation proposal
- Different types of arguments
- Each argument type defines preconditions for its usage. If the preconditions are met, then the agent may use the argument.
- The agent needs a strategy to decide which argument to use
- Most of the times assumes a BDI model

72

- **Appeal to past promise** - the negotiator *A* reminds agent *B* of a past promise regarding the NO, i.e., agent *B* has promised to the agent A to perform or offer NO in a previous negotiation.
- ***Preconditions***: *A* must check if a promise of NO (future reward) was received in the past in a successfully concluded negotiation.

- **Promise of a future reward** - the negotiator *A* promises to do a NO for the other agent *A* at a future time.
- ***Preconditions***: *A* must find one desire of agent *B* for a future time interval, if possible a desire which can be satisfied through an action (service) that *A* can perform while *B* can not.

- **Appeal to self interest** - the agent *A* believes that concluding the contract for NO is in the best interest of *B* and tries to persuade *B* of this fact.
- ***Preconditions***: *A* must find (or infer) one of *B* desires which is satisfied if *B* has NO or, alternatively, *A* must find another negotiation object NO' that is previously offered on the market and it believes NO is better than NO'.

- **Threat** - the negotiator makes the threat of refusing doing/offering something to *B* or threatens that it will do something to contradict *B*'s desires.
- ***Preconditions***: *A* must find one of *B*'s desires directly fulfilled by a NO that *A* can offer or *A* must find an action that is contradictory to what it believes is one of *B*'s desires.

References

- T.W. Sandholm. Distributed rational decision making. In Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, G. Weiss (Ed.), The MIT Press, 2001, p.201-258.
- M. Wooldrige. An Introduction to MultiAgent Systems, John Wiley & Sons,2002.
- J.S. Rosenschein, G. Zlotkin. Designing conventions for automated negotiation. In Readings in Agents, M. Huhns & M. Singh (Eds.), Morgan Kaufmann, 1998, p.253-370.
- M.P. Wellman. A market-oriented programming environment and its applications to distributed multicommodity flow problems. Journal of Artificial Intelligence Research, 1, 1993, p.1-23.
- N.R. Jennings, e.a., Automated negotiation: prospects, methods, and challenges, Journal of Group Decision and Negotiation, 2000.
- S. Kraus, K. Sycara, A. Evenchik, Reaching agreements through arumentation: a logical model and implementation, Artificial Intelligence, Elsevier Science, 104, 1998, p. 1-69.
- A. Florea, B. Panghe. Achieving Cooperation of Self-interested Agents Based on Cost", In Proceedings of the 15th European Meeting on Cybernetics and System Research, Session: From Agent Theories to Agent Implementation, Vienna, 2000, p.591-596.