UNIT–4 : Software Agents (PYQ Solution)

AKTU · Artificial Intelligence (BCAI501) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. Define rational agent. (Jan 2026)

A rational agent is an agent that, for every possible percept sequence, selects the action that is expected to maximise its performance measure, given the evidence from its percepts and its built-in knowledge. Rationality depends on the performance measure, the agent's prior knowledge, the actions available and the percept history. Rational does not mean omniscient or perfect; it means making the best decision with the information available. A thermostat or a chess program is a simple rational agent.

Q2. Define negotiation in the context of software agents. (Jan 2025)

Negotiation is a process in which two or more software agents with different, possibly conflicting goals communicate and exchange proposals to reach a mutually acceptable agreement. It involves a negotiation protocol (rules of interaction), a strategy (how each agent bids or concedes) and an objective such as price or resource share. Examples are agents negotiating the price of a product, or sharing a bandwidth. Common forms are auctions, bargaining and the contract net protocol.

Q3. Define speech act theory. How will you define the speech act if performative = request and content = "the door is closed"? (2023)

Speech act theory (Austin, Searle) treats communication as action: when agents speak, they perform acts such as requesting, informing, promising or ordering, not just transmit data. Each message has a performative (the type of act) and a content (the proposition).
Here performative = request, content = "the door is closed", so the speech act

is request("the door is closed"): the sender asks the receiver to bring about the state in which the door is closed, i.e. "Please close the door."

Q4. Differentiate between reactive agent and deliberative agent. (2023)

A reactive agent responds directly to the current percept using condition-action rules, with no symbolic world model or planning; it is fast and robust but short-sighted, e.g. a thermostat. A deliberative agent keeps a symbolic model of the world and reasons and plans to achieve goals before acting; it is flexible and goal-directed but slower and needs accurate knowledge, e.g. a route-planning robot. Hybrid agents combine both layers.

Q5. What are the key characteristics that define an intelligent agent in a multi-agent system? (2024)

The key characteristics are:

- **Autonomy: operates without direct human control and controls its own actions.**
- **Reactivity: perceives the environment and responds in time to changes.**
- **Pro-activeness: takes initiative to achieve its goals.**
- **Social ability: communicates, cooperates and negotiates with other agents.**
- **Also learning, mobility and rationality.**
  Together they allow agents to work as a team to solve problems that a single agent cannot.

Q6. Explain the importance of communication among intelligent agents in a multi-agent system. (2024)

Communication lets agents share information, which is essential because each agent has only partial knowledge. It enables coordination of actions, avoiding conflicts and duplicated work, and cooperation towards common goals. Through negotiation and auctions agents resolve conflicts of interest and allocate tasks and resources. It also supports distributed problem solving, trust building and flexibility. Standard languages such as KQML and FIPA-ACL make communication possible among heterogeneous agents.

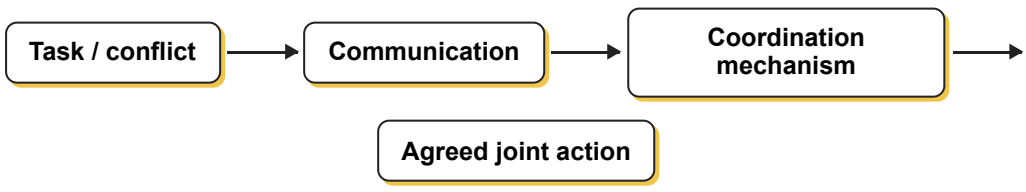
SECTION B Answers (≈250 Words Each)

Q1. Explain the coordination mechanisms used in multi-agent systems. (Jan 2026)

Coordination is the management of interdependencies among the activities of agents so that they work together efficiently and avoid conflicts. It is needed because agents have limited knowledge, share resources and have interdependent goals.

Main mechanisms

- **Organisational structuring:** roles, authority and fixed responsibilities are predefined (hierarchy, market, team). Master-slave is a simple example.
- **Contracting (Contract Net Protocol):** a manager announces a task, agents bid, and the best bid gets the contract.
- **Multi-agent planning:** agents build a joint plan (centralised) or merge individual plans (distributed) so that actions do not conflict.
- **Negotiation and auctions:** agents exchange offers to allocate resources and tasks (English, Dutch, Vickrey auctions).
- **Blackboard / shared memory:** agents post partial results to a common data space (indirect coordination).
- **Social laws and conventions:** agreed rules such as "keep left" prevent conflict without messages.
- **Partial Global Planning (PGP):** agents exchange parts of their plans and adjust them.
- **Voting and consensus:** group decisions by preference aggregation.
- **Stigmergy:** indirect coordination through changes in the environment, like ants leaving pheromones.



Coordination in a multi-agent system

Mechanism	Type	Example
Organisational structure	Pre-defined	Warehouse robots with roles
Contract net	Negotiated	Task allocation
Blackboard	Indirect	Speech understanding
Social laws	Convention	Traffic rules
Auction	Market	Ad slots

Conclusion: the choice of mechanism depends on how dynamic the system is and how much communication it can afford; together they give coherent behavior of the agent group.

Q2. Describe the negotiation process in multi-agent systems with examples. (Jan 2025)

Negotiation is the process by which agents with conflicting interests reach an agreement through exchange of proposals. It is a key tool of cooperation and resource sharing among **self-interested** agents.

Components

- **Negotiation set:** the possible proposals the agents may agree on.
- **Protocol:** the rules, who can say what and when (e.g. alternating offers).
- **Strategy:** the private decision rule used by each agent to make offers and concessions.
- **Utility functions:** the value each agent gives to each outcome.
- **Reservation value:** the worst deal the agent will accept.

Process (steps)

1. **Initiation:** one agent proposes a deal or announces a task.
2. **Proposal / offer:** agents send offers with price, quantity or time.
3. **Evaluation:** each receiver computes the utility of the offer.
4. **Counter-proposal or concession:** if unacceptable, it replies with a modified offer.
5. **Agreement or termination:** the process ends when an offer is accepted or all options are exhausted (**conflict deal**).



Negotiation cycle

Examples

- **E-commerce:** a buyer agent wants a laptop for Rs 40,000; the seller agent asks Rs 50,000. Offers move 40,000 → 49,000 → 42,000 → 46,000 → 44,000 and they settle at Rs 44,000.
- **Meeting scheduling:** personal agents propose slots and negotiate until all calendars match.
- **Task allocation (contract net):** bidding agents quote cost and time; the manager awards the contract.
- **Auctions:** English (rising bids), Dutch (falling price), Vickrey (second-price) are one-to-many negotiations.

Desirable properties: efficiency (Pareto optimal), stability, simplicity, distribution and low communication cost.

Conclusion: negotiation lets autonomous agents find agreements that benefit both sides without any central controller.

Q3. Define the two types of agent communication language (ACL). Explain in detail with some examples. (2023)

An **Agent Communication Language (ACL)** is a standard language in which agents exchange messages so that they understand each other irrespective of their internal design. It is based on **speech act theory**: each message has a **performative** and a **content**. The two best known ACLs are **KQML** and **FIPA-ACL**.

1. KQML (Knowledge Query and Manipulation Language)

- Developed by the DARPA Knowledge Sharing Effort (early 1990s).
- Message layers: **content** (the actual knowledge), **message** (performative and parameters such as :sender, :receiver, :language, :ontology), and **communication** (transport details).
- Performatives: ask-one, ask-all, tell, achieve, subscribe, advertise, reply.
- Example:

```
(ask-one :sender A1 :receiver Stock-Server
         :language PROLOG :ontology NYSE
         :reply-with ibm-stock
         :content (price ibm ?price))
```

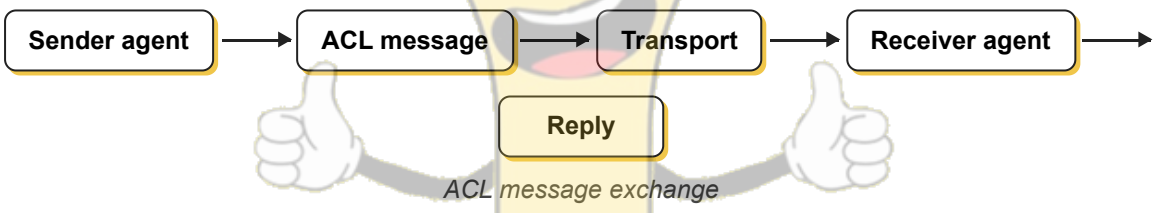
- Reply: `(tell :sender Stock-Server :receiver A1 :in-reply-to ibm-stock :content (price ibm 14))`.
- Weakness: informal semantics, many dialects.

2. FIPA-ACL (Foundation for Intelligent Physical Agents)

- Similar syntax but with a **formal semantics** defined in a modal logic of belief, desire and intention (feasibility preconditions and rational effects).
- About 22 communicative acts: inform, request, query-if, query-ref, propose, accept-proposal, reject-proposal, agree, refuse, cfp (call for proposal), subscribe, confirm.
- Example:

```
(inform :sender agent-i :receiver agent-j
        :content "weather(today, rainy)"
        :language Prolog :ontology weather-ontology)
```

- Supports interaction protocols such as Contract Net and auctions.



ACL message exchange

Feature	KQML	FIPA-ACL
Origin	DARPA	FIPA
Semantics	Informal	Formal (BDI logic)
Performatives	About 36	About 22
Standardisation	None	IEEE standard
Protocols	Few	Many defined

Conclusion: both ACLs give agents a common way to ask, tell and request things, but FIPA-ACL is the more precise and standardised option.

Q4. How do intelligent agents perceive and act within their environment in the context of multi-agent systems? (2024)

In a **multi-agent system (MAS)** several agents share one environment. Each agent continuously runs a **perceive-decide-act cycle**.

Perception

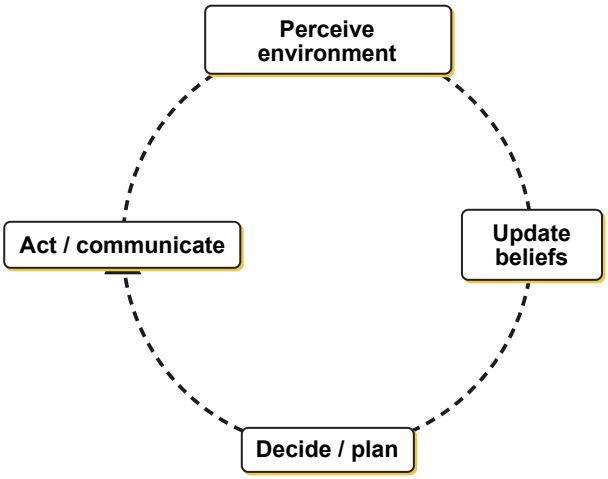
- Agents use **sensors** (physical or software) to get **percepts** of the environment: the state of resources and the effects of other agents' actions.
- Perception is usually **partial** and local, so agents keep an **internal state or beliefs** (e.g. BDI: beliefs).
- Information may also come by **messages** from other agents, which extends perception.

Decision

- The agent program maps percepts and internal state to an action using rules, planning, learning or utility maximisation.
- It must take into account the **goals and likely actions of other agents** (strategic reasoning, game theory).

Action

- Agents use **actuators** (or software commands) to change the environment, move, or **send messages**.
- Actions are of two kinds: **physical/domain actions** and **communicative actions** (speech acts).
- Because actions of several agents may happen together, the effect on the environment is the **joint result**; conflicts can arise, so coordination is needed.



Perceive-decide-act cycle in MAS

Features of the MAS environment: dynamic, partially observable, non-deterministic as seen by one agent (others act), often continuous and distributed.

Example: warehouse robots perceive shelves and positions through cameras and messages, decide routes, then move and announce their paths so that other robots avoid collisions.

Conclusion: agents in a MAS perceive through sensors and messages and act through physical and communicative actions, coordinating to deal with other agents.

SECTION C Answers (≈500 Words Each)

Q1. Explain the complete architecture of intelligent agents including deliberative, reactive and hybrid layers. (Jan 2026)

An **agent architecture** is the software design that decides how percepts are turned into actions. There are three main families.

1. Deliberative (symbolic) architecture

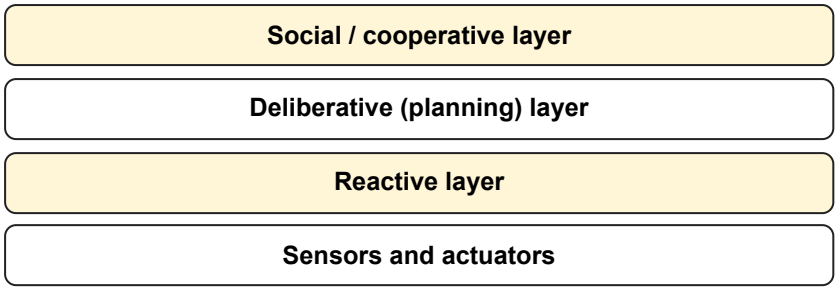
- Keeps an explicit **symbolic model** of the world and uses logical reasoning and **planning** to choose actions.
- Example: the **BDI (Belief-Desire-Intention)** model. **Beliefs** are what the agent thinks is true; **desires** are goals; **intentions** are the goals it has committed to. The cycle is: update beliefs, generate options, filter to intentions, execute plan.
- Strengths: goal-directed, flexible, explainable. Weaknesses: slow, needs complete accurate models, the **transduction and representation problems**.

2. Reactive architecture

- No symbolic model or search; behavior emerges from simple **situation-action rules**.
- Example: Brooks' **subsumption architecture** with layered behaviors (avoid obstacles, wander, explore) in which lower layers can **suppress or override** higher ones.
- Strengths: very fast, robust, simple. Weaknesses: short-term, no long-term planning, hard to design for many behaviors.

3. Hybrid (layered) architecture

- Combines a fast reactive layer with a deliberative layer.
- **Horizontal layering:** every layer sees the percepts and suggests actions, and a **mediator** decides (e.g. TouringMachines). Simple but needs control among layers.
- **Vertical layering:** percepts enter the lowest layer and move up; actions come out from the top (one pass) or return down (two pass) (e.g. InteRRaP).



Hybrid layered agent architecture

Typical hybrid layers (InteRRaP style)

- **Reactive (behavior-based) layer:** immediate responses, e.g. avoid a collision.
- **Local planning layer:** plans to achieve own goals.
- **Cooperative planning layer:** deals with other agents, negotiation and joint plans.
- A **control/mediator** passes control up when the lower layer cannot cope and handles conflicts.

Feature	Deliberative	Reactive	Hybrid
World model	Symbolic, explicit	None	Partial
Speed	Slow	Very fast	Fast for urgent events
Planning	Yes	No	Yes
Robustness	Low	High	High
Example	BDI	Subsumption	TouringMachines, InteRRaP

Conclusion: hybrid architectures combine the quick reactions of reactive layers with the long-term intelligence of deliberative layers, and are used in robotics and autonomous vehicles.

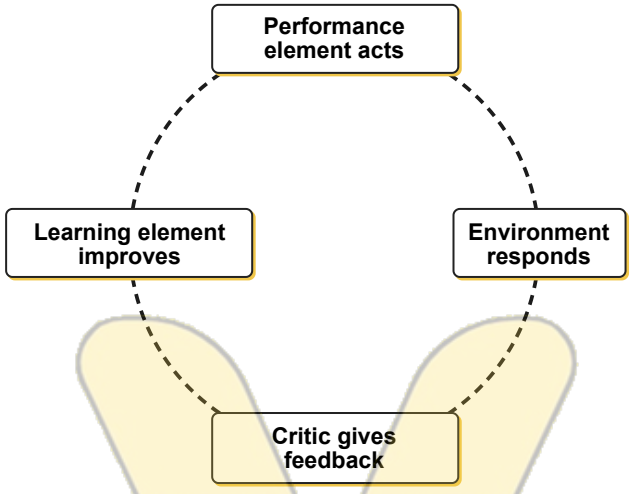
Q2. Describe how learning agents adapt their behavior over time. (Jan 2026)

A **learning agent** improves its performance with experience, so it can work in **initially unknown environments** and become better than its initial knowledge allows.

Components

- **Performance element:** selects external actions (the agent program of other types).

- **Critic:** compares the agent's behavior with a fixed **performance standard** and gives **feedback** (reward or penalty).
- **Learning element:** uses the critic's feedback to **modify the performance element** (rules, model, utility estimates).
- **Problem generator:** proposes **exploratory actions** that give new experience, even if they are not the best at present.



Learning agent loop

How adaptation happens

- **Supervised learning:** learns from labelled examples (e.g. a spam filter updates from marked emails).
- **Unsupervised learning:** discovers patterns and clusters in data.
- **Reinforcement learning (RL):** the agent tries actions, receives **rewards**, and learns a **policy** that maximises long-term reward. The update of Q-learning is: $Q(s,a) \leftarrow Q(s,a) + \alpha\,[r + \gamma \max Q(s',a') - Q(s,a)]$.
- **Exploration vs exploitation:** the agent balances trying new actions and using known good ones (ε-greedy).
- **Model learning:** it improves its model of how the environment responds, which gives better planning.
- **Online / continuous learning:** parameters are updated as new data arrive, so the agent follows **changes** in the environment (concept drift).
- **Multi-agent learning:** agents learn from others' behavior and adapt strategies.

Example: self-driving car or game agent. The car begins with basic rules; the critic penalises hard braking or collisions and rewards smooth, safe trips; the learning element changes its driving policy; the problem generator tries new lane-change

times in simulation. Over time the car drives better and adapts to new roads and traffic.

Part	Function
Performance element	Choose action
Critic	Evaluate result
Learning element	Change behavior
Problem generator	Explore

Challenges: needs a lot of data or trials, risk in exploration, credit assignment (which action caused the reward), and stability when many agents learn together.

Conclusion: through feedback and exploration, learning agents gradually update their knowledge and policies and so adapt to new situations.

Q3. Explain how argumentation is used for resolving conflicts in multi-agent systems. (Jan 2025)

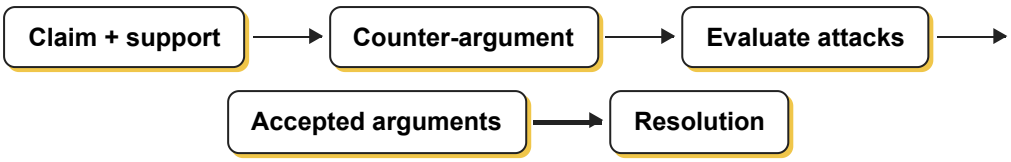
Argumentation is a form of reasoned communication in which agents exchange **arguments**, that is **claims supported by reasons**, to persuade others or to resolve inconsistencies. Unlike simple bargaining, agents can give **justifications**, criticise opponents' claims and change **beliefs or preferences**.

Structure of an argument

- **Claim (conclusion):** what is to be accepted.
- **Support (premises):** evidence and rules leading to the claim.
- **Attack:** one argument can attack another by **rebutting** (contradicting the claim), **undermining** (attacking a premise) or **undercutting** (attacking the rule that connects them).

Process

1. An agent proposes a position with an argument.
2. Other agents examine it and respond with **counter-arguments** or request clarification.
3. The agents decide which arguments are **acceptable** (not defeated).
4. They revise beliefs or goals and reach agreement, or end in agreed disagreement.



Argumentation dialogue

Formal models

- **Dung's abstract argumentation framework:** a set of arguments with an **attack relation**; semantics (grounded, preferred) compute which arguments are accepted.
- **Dialogue games:** rules for moves (assert, challenge, retract) in persuasion, negotiation or inquiry dialogues.
- **Preference-based argumentation:** attacks are weighed by agent preferences and trust.

Conflict types resolved: disagreement over facts (beliefs), over plans or goals, and over resource allocation.

Example: Agent A wants to use a shared printer now. B objects: "My job is urgent: deadline in 10 minutes." A counters: "My job has a higher priority by company policy." B presents the manager's override. A accepts that B's argument is stronger, and B prints first.

Benefits: explanations, better outcomes than blind bargaining, can change opponent's preferences, transparency.
Limitations: complex computation, needs shared argument languages, risk of manipulation.

Conclusion: argumentation lets agents settle conflicts through reasons rather than power or random choice, giving more rational and explainable agreements.

Q4. Discuss the use of trust metrics in multi-agent decision-making. (Jan 2025)

In open multi-agent systems, agents are **autonomous and may be unreliable or dishonest**, so an agent must decide whom to depend on. **Trust** is an agent's belief in another's ability and willingness to act as promised. **Trust metrics** are numerical or symbolic measures of this belief.

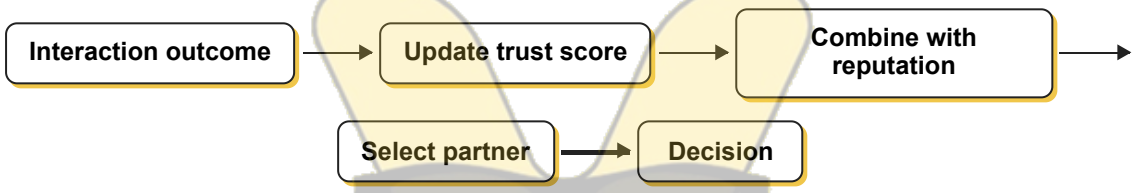
Sources of trust information

- **Direct experience:** the outcome of earlier interactions.

- **Witness / reputation:** opinions of third parties, collected as a **reputation** score.
- **Certification and organisational roles:** credentials from a trusted authority.
- **Stereotypes / reasoning** about the other agent's properties.

Common metrics and models

- **Average success rate:** trust = successful interactions / total interactions.
- **Beta reputation:** trust = (s + 1) / (s + f + 2) based on s successes and f failures.
- **Weighted combination** of direct trust and reputation: T = w × T_direct + (1 − w) × T_reputation.
- **Time-decay** gives more weight to recent behavior.
- Models: **REGRET, FIRE, TRAVOS, EigenTrust** (for peer-to-peer networks), **Marsh's** model.
- Dimensions: **competence, honesty, availability, quality**.



Trust in decision making

Use in decision making

- **Partner selection:** choose the supplier or contractor with the highest expected trusted utility.
- **Risk assessment:** expected utility = trust × benefit − (1 − trust) × loss.
- **Contract net and auctions:** weigh bids by the trust of the bidder.
- **Coalition formation:** agents join groups with trusted members.
- **Information filtering:** ignore data from low-trust sources and detect **malicious agents** (collusion, fake ratings).
- **Delegation:** trusted agents get tasks of higher importance.

Example: an e-commerce buyer agent has three seller agents. Seller 1 has 18 successes in 20 sales, trust ≈ 0.86; Seller 2 has 5 in 10, trust ≈ 0.5. The buyer prefers Seller 1 even at a slightly higher price.

Challenges: cold start for new agents, false reviews, subjectivity, privacy.

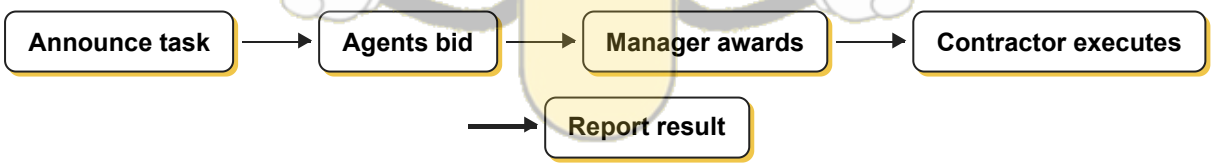
Conclusion: trust metrics reduce risk and improve cooperation and decisions in multi-agent systems by predicting how reliable other agents will be.

Q5. Explain in detail the contract net protocol used for communication between multi-agent systems. Also explain each stage of the protocol. (2023)

The **Contract Net Protocol (CNP)**, proposed by **Reid Smith (1980)**, is a **task-allocation protocol** for distributed problem solving, modelled on how a contractor gives out jobs. Agents take two roles: a **manager** (announces a task) and **contractors** (agents that may carry it out). The same agent can be both.

Stages of the protocol

1. **Task announcement (call for proposals):** the manager breaks a problem into tasks and **broadcasts** a task announcement with its description, **eligibility conditions**, bid deadline and required information.
2. **Bidding:** interested agents that meet the conditions evaluate the task against their capability and load, and send a **bid** (cost, time, quality). Others send **refuse** or stay silent.
3. **Awarding (evaluation):** the manager compares the bids with an evaluation function and **awards the contract** to the best contractor; it sends **reject** messages to the others.
4. **Expediting (execution):** the contractor performs the task, perhaps **sub-contracting** parts to other agents, and keeps the manager informed.
5. **Result reporting:** the contractor sends the **result** (inform-done or failure) to the manager, who may issue payment or update the contractor's reputation.



Contract net protocol stages

Messages in FIPA CNP: cfp, propose, refuse, reject-proposal, accept-proposal, inform, failure.

Example: a delivery manager agent announces "deliver 10 parcels to Zone 3 by 5 pm". Truck agents bid with costs Rs 400, Rs 350 and Rs 500; the manager awards the contract to the Rs 350 truck, which delivers and reports completion.

Advantages: flexible, dynamic load balancing, no central knowledge needed, handles failures by re-announcing the task, natural for hierarchical decomposition.
Disadvantages: broadcast communication overhead, manager is a bottleneck, assumes agents are honest, and bidders may be busy when the award arrives.

Stage	Message	Sender
Announcement	cfp	Manager
Bidding	propose / refuse	Contractors
Award	accept / reject	Manager
Execution result	inform / failure	Contractor

Conclusion: CNP efficiently allocates tasks among autonomous agents through announcement, bidding, award, execution and reporting.

Q6. Describe argument, negotiation and bargaining in terms of multi-software-agent systems. What role does bargaining play in resolving conflicts and reaching agreements among intelligent agents? (2024, 2023)

In a multi-agent system, agents have different goals and limited resources, so **conflict** is normal. Three related mechanisms are used to resolve it.

(a) Argument (argumentation)

- Agents exchange **claims with supporting reasons** and counter-arguments, aiming to **persuade** and change beliefs or preferences.
- Example: "Use route B because route A has an accident (report #5)." Accepted if the evidence is stronger.

(b) Negotiation

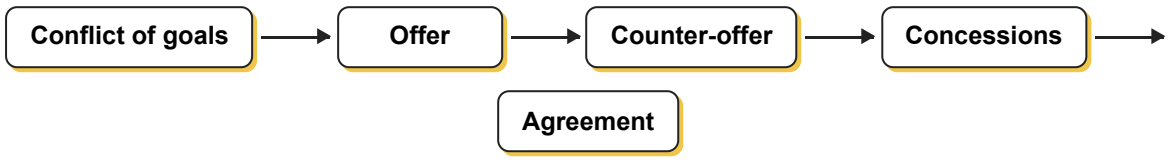
- A general process in which agents **exchange proposals** under a protocol until they **agree**. It includes bargaining, auctions, contract net and argument-based negotiation.
- Elements: negotiation set, protocol, strategy, utilities.
- Example: buyer and seller agents agree on price and delivery time.

(c) Bargaining

- A **specific form of negotiation** between two (or few) agents on **how to divide a surplus** or fix terms such as price, where each side wants a larger share.
- Typical model: **alternating offers** (Rubinstein). Agent 1 proposes a split, agent 2 accepts or counter-offers; delay costs both through a **discount factor**, which

pushes them to settle. The **reservation value** is the worst outcome each will accept.

- **Zone of agreement:** exists if the buyer's maximum price ≥ seller's minimum price.
- Solution concepts: **Nash bargaining solution** (maximises the product of the utility gains), Pareto optimality, subgame-perfect equilibrium.



Bargaining process

Role of bargaining in conflict resolution

- Gives a **structured, fair and efficient** way to reach agreement without a central authority.
- Allows each agent to **trade off** its goals (concede on less important issues).
- Prevents deadlock and loss from continuing conflict because delay is costly.
- Supports **resource allocation, price fixing and task scheduling** in e-commerce and supply chains.
- Agreements found can be **Pareto efficient**, so no agent can gain without another losing.

Example: two cloud agents share 100 units of bandwidth. A first asks for 70, B offers 40. A moves to 60, B to 50; with each making concessions they settle at 55:45 before the deadline.

Mechanism	Basis	Best for
Argumentation	Reasons	Resolving belief or goal conflict
Negotiation	Proposals under protocol	General agreement
Bargaining	Offers and counter-offers	Dividing a surplus / price

Conclusion: argument convinces, negotiation is the general framework, and bargaining is the practical give-and-take that settles conflicts between self-interested agents.

Q7. What are the different communication paradigms used by intelligent

agents, and how do they facilitate collaboration? (2024)

Agents must exchange information to collaborate, and they use several **communication paradigms**.

1. Message passing (direct communication)

- Agents send messages point-to-point (or multicast) using an **Agent Communication Language** such as **KQML or FIPA-ACL** with performatives like inform, request, propose.
- Based on **speech act theory**. It is flexible but needs shared ontologies and addresses.

2. Blackboard (shared memory) communication

- Agents read and write a **common data structure** (blackboard) without knowing each other. A **controller** decides which agent acts.
- Good for cooperative problem solving; the blackboard may become a bottleneck.

3. Broadcast / publish-subscribe

- A sender announces to many agents or to a topic; interested agents subscribe. Used in the contract net announcement.

4. Mediated communication (facilitators / brokers)

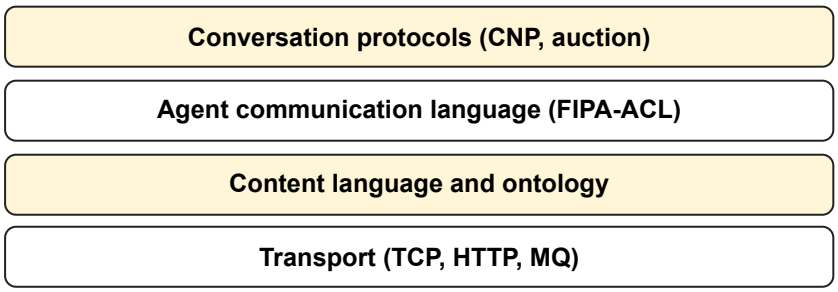
- A **middle agent** such as a broker, matchmaker or yellow-pages agent helps locate agents and routes messages. It decouples agents and supports open systems.

5. Indirect communication (stigmergy)

- Agents change the **environment** and others sense it, as ants do with pheromones. Needs no direct messages and scales well.

6. Negotiation and dialogue protocols

- Structured conversations such as the contract net, auctions and argumentation.



Layers of agent communication

How they facilitate collaboration

- **Information sharing** to overcome partial knowledge.
- **Task allocation** and coordination (who does what, when).
- **Conflict resolution** through negotiation and argument.
- **Synchronisation** of actions and teamwork.
- **Scalability and openness**, since agents of different makers can interoperate.

Paradigm	Coupling	Example
Message passing	Direct	FIPA-ACL inform
Blackboard	Shared space	Speech understanding
Broadcast / pub-sub	Group	Task announcement
Broker	Mediated	Yellow pages
Stigmergy	Environment	Ant routing

Conclusion: each paradigm suits a different level of coupling, and together they let agents share knowledge, coordinate and work as a team.

