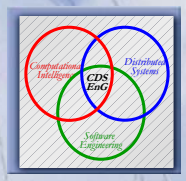
ES595a - Advanced Topics in Software & Systems Design
Cooperative Distributed Systems Engineering: Technologies &Applications



W03: *Agent-Orientation*

Western

Outlines

- AO vs. Others
- AO: Coordinated, Intelligent Rational Agent

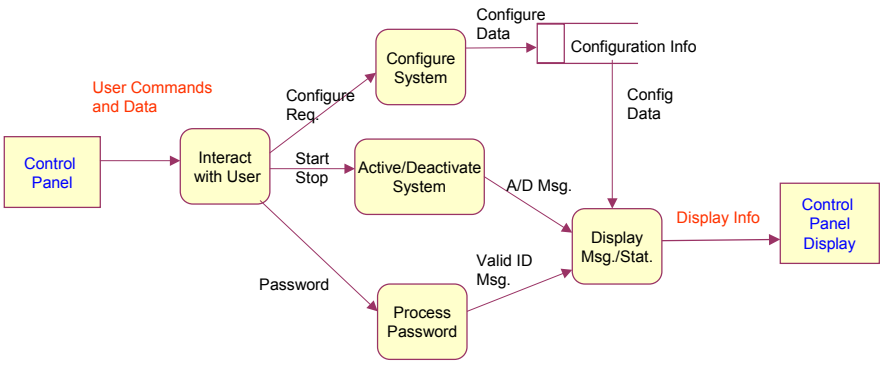
  CIR-Agent Model

Functional Oriented

- The system functionality is described in terms of

  data transformation process

Functional Oriented Approach

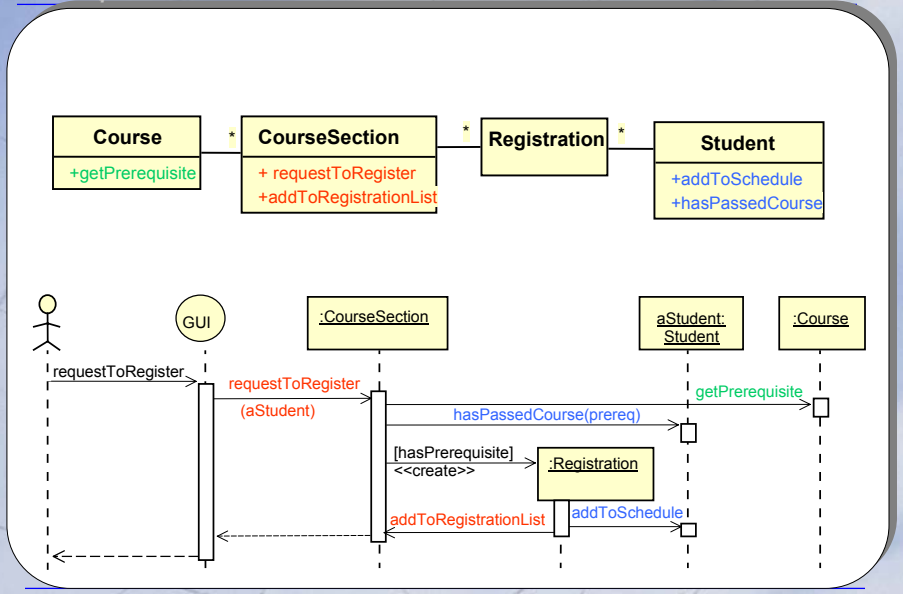


Object Oriented Paradigm

- The system functionality is described in terms of

A set of interoperating objects

Sequence Diagram
Example



Agent-Orientation –Ghenniwa's

- Agent-orientation is the next generation for
  - software engineering paradigms
  - programming methodologies, and
  - computational modeling,

CDS: Design Concepts^{++} Recall…

- Abstraction
- Refinement
- Modularity
- Information Hiding
- Interface
- Functional Independence
- Architecture
- Reusability

- Autonomy
- Coordination
  - Communication
  - Interaction
- Cooperation
- Adaptability

Agent View–Ghenniwa's

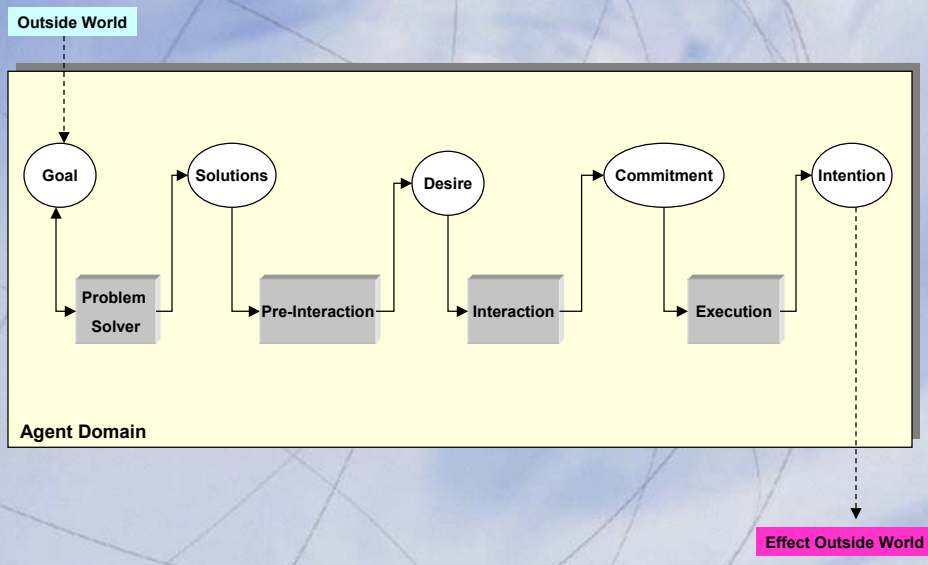
- A level of abstraction to construct a computational systems (agents)
  - that inherent the agenthood features including:
    - Primary
      - coordination and rationality
    - Secondary
      - intelligence and learning

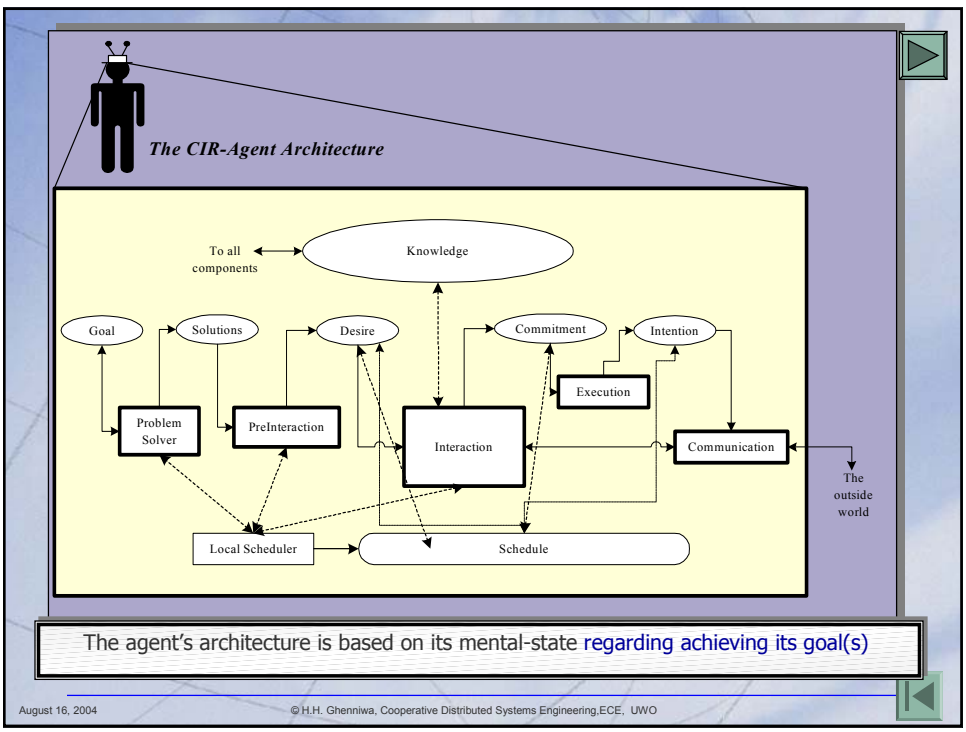
They are useful for distributed computation in open environments

The Agent (System)–Ghenniwa's

- An agent as an artifact is an individual collection of primitive components
  - Each component is associated with a particular functionality supports a specific agent's mental state as related to its goal
    - A particular arrangement of the components is required to constitute an agent
      - This arrangement reflects the pattern of the agent's mental state as related to its reasoning to achieve a goal
    - The components can be structured to include:
      - knowledge
      - Capabilities
        - problem-solving
        - communication
        - interaction

Agent's Mental State



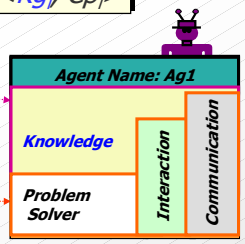


The agent's architecture is based on its mental-state regarding achieving its goal(s)

The CIR-Agent Model

- An agent is an entity which possesses knowledge and capabilities;

$Ag_i = <Kg_i, Cp_i>$



- Knowledge (Kg_i)
  - domain-dependent
  - self-model
  - others-models
- Capabilities (Cp_i)
  - Reasoning
    - problem-solving a
    - interaction
  - Communication

The Mental-State

- the agent's internal structural representation, at a given instance of time, for
  - Goal: goal-; solution-; desire-; commitment-; intention-state
  - Reasoning activities toward achieving a goal:
    - Problem-solving; pre-interaction; interaction; execution
  - Time
    - Future: virtual-time line (*local-schedule*)
    - Current: real-time clock
    - Past: virtual-time line associated with local-view (*local-history*)
  - Local view of the world
  - Coordination knowledge
    - beliefs related to the interdependency problems

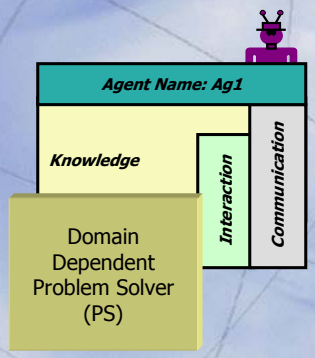
Implementation Assumptions!

- No specific assumptions need to be made on the detailed design of the agent components
  - Therefore, the components can be developed and implemented using object oriented or functional oriented approach,

provided
the designer conceptualizing a specific architecture of the agent

  - In other words, object oriented technology for example, can be used to *enable* agent-based technology,
    - but agenthood features such as coordination and cooperation required by agents is not currently supported within OO technology

The Problem-Solver: Domain Dependent



What is a problem solver?

PS ∈ {Car Assembly, Information Gathering, Object-Tracker,… }

The Problem-Solver

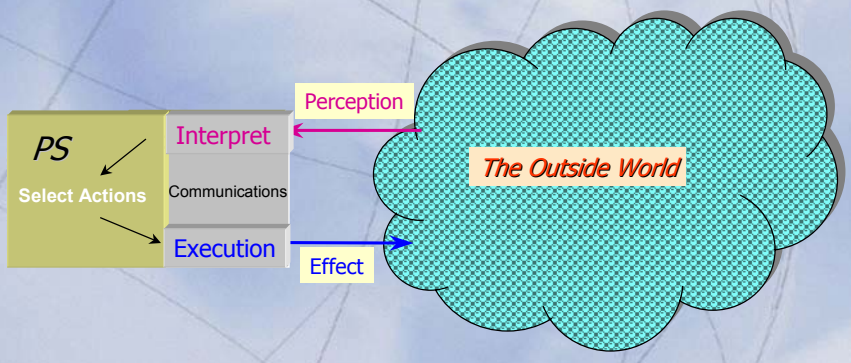
- Goal-driven approach
  - more appropriate for cooperative distributed systems

$$PS: G_i \; X \; Ac_i \; X \; W^t_i \rightarrow S_i^g$$

  - G_i: set of Ag_i's goals
  - AC_i: set of Ag_i's domain actions
  - W_i: set of Ag_i's world history
  - S_i^g: set of Ag_i's solutions for $g \in G_i$

- Ps: local-control over achieving a domain goal
  - irresponsible for domain-actions associated with interdependency problem

PS ←→ The Outside World



PS Autonomy

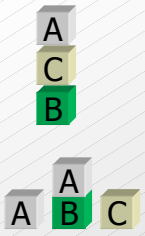
1. Perception: <Action-descriptor>
   - *Action=<Action-Descriptor, Action-Body>*
     - Static, Closed-World Assumption
     - PS: Identify Action-body
2. Perception: <State, Action-descriptor>
   - *Action=<Action-Descriptor, Preconditions, Action-Body>*
     - Dynamic, Closed World Assumption
     - PS: Identify Action-Body

       React to the current state of World
3. Perception: <State, Goal>
   - *Action=<Action-Descriptor, Preconditions, Action-Body, Postconditions>*
     - Dynamic, Open Environment
     - PS: Identify Possible Course of Actions to achieve Goal

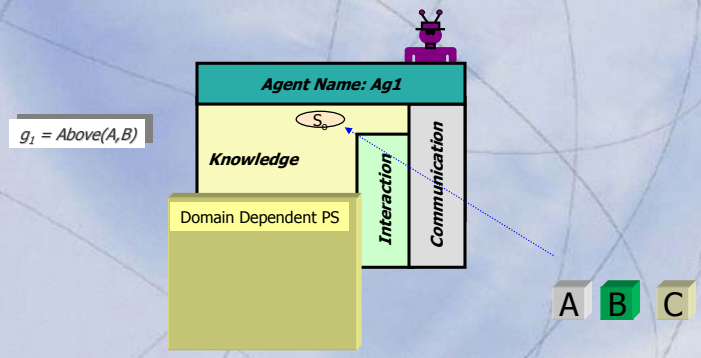
PS → Block World Domain

- Problem Solving:

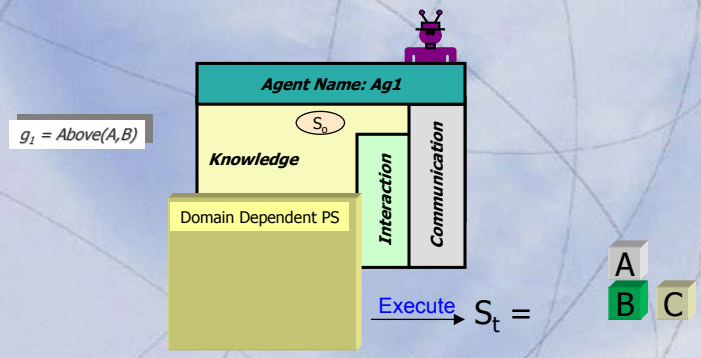
  the *domain-dependent role* of the agent
- Representation
  - e.g., state-based model
    - *World: <Domain-Objects, Relationships>*
    - *Action: State → State*
    - Example
      - S_1 = *on(A,C)*; S_2 = *On(C,B)*
      - *Actions*: a_1 =*Puton(A,B);*



PS → Block World Domain
Single Goal

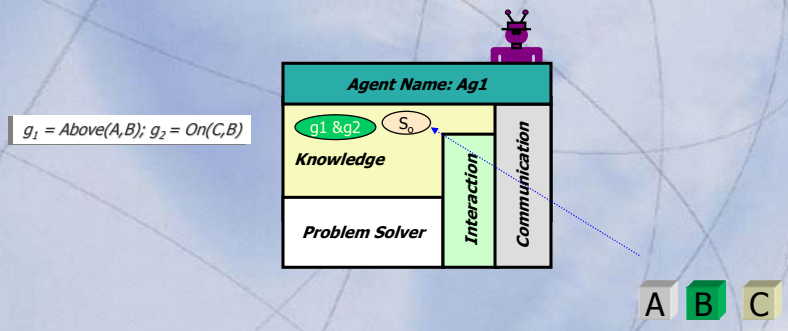


PS → Block World Domain
Single Goal



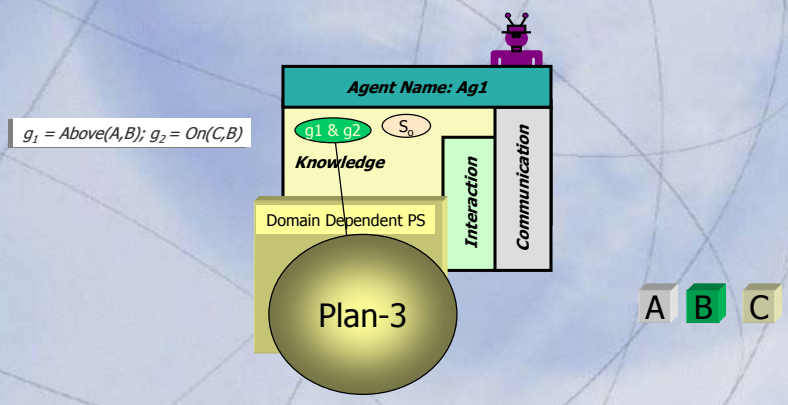
PS → Block World Domain

Multiple Goals



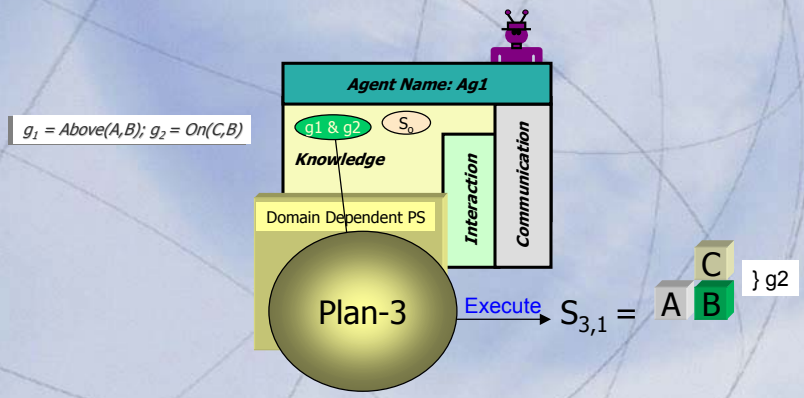
PS → Block World Domain

Multiple Goals



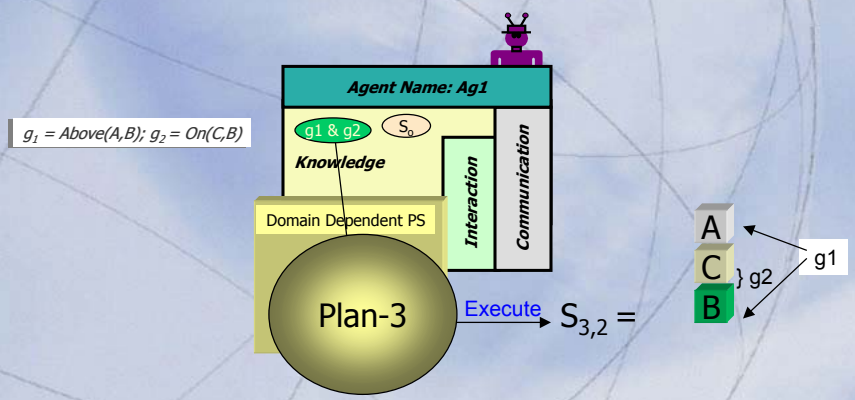
PS → Block World Domain

Multiple Goals



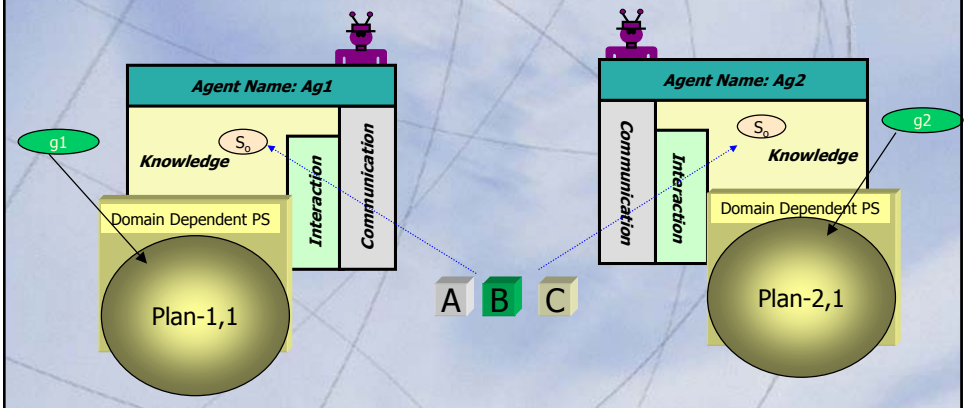
PS → Block World Domain

Multiple Goals



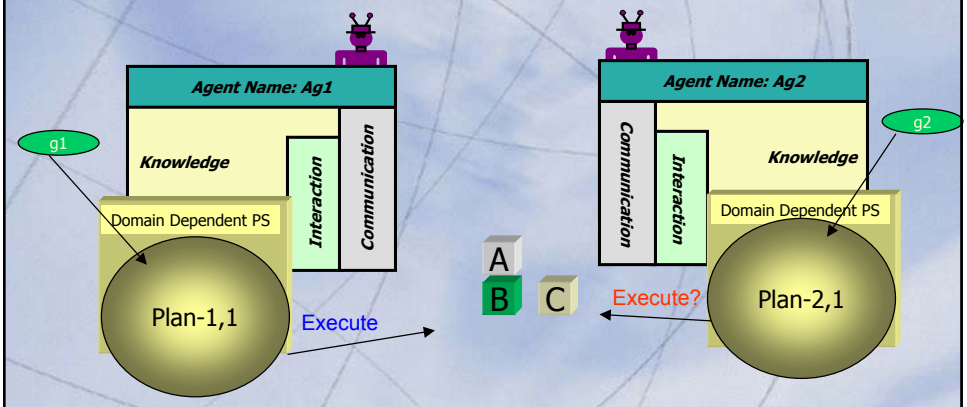
PS → Multiple Agents

g_1 = Above(A,B); g_2 = On(C,B)

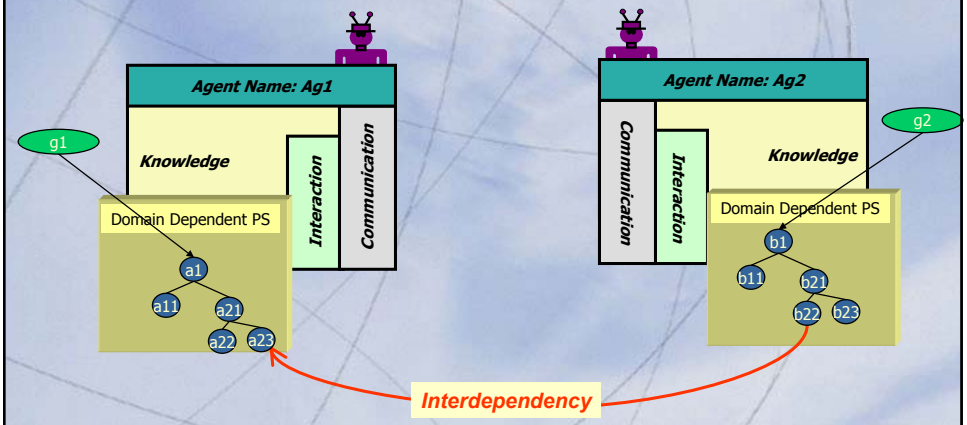


PS → Multiple Agents

g_1 = Above(A,B); g_2 = On(C,B)



PS → Multiple Agents Interdependency…



Interdependency –Ghenniwa's

- Interdependencies are goal-relevant relationships among actions performed by various agents
  - For instance, Ag_1 attempts to achieve a goal that is beyond its capability, but it can be achieved with the help of Ag_2
  - Another kind interdependency that may exist between the agents is when each has a goal that can only be achieved through the use of a `shared' resource (e.g., a printer in the Lab)

Modeling Interdependency Types–Ghenniwa's

Interdependency types Include:

- Capability interdependencies
  - Decomposition interdependencies
- Interest interdependencies
  - Conflict
  - Common
  - Simultaneous
  - Contradict
- Resource interdependencies
- Knowledge interdependencies

Modeling Interdependency –Ghenniwa's

- Interdependency Types
  - Capability interdependencies
    - are related to the agent's bounded capability of achieving goals
    - Decomposition interdependencies
      - are related to the agent's decomposition of goals into sub-goals, where achieving some of these sub-goals might be beyond its capability ---non self-containment.
  - Knowledge interdependencies
    - are related to the required knowledge items of the domain actions that might be affected by other agents.

Modeling Interdependency *(Cont.)*

- Interdependency Types *(cont.)*
  - Interest interdependencies
    - are related to the types of interrelationships that exist between any pair of different agents' goals.
    - These interrelationships include
      - contradict
      - conflict
      - common
      - simultaneous
  - Resource interdependencies
    - are related to the agents' domain actions that require shared resources
- Uncertainty about Interdependencies