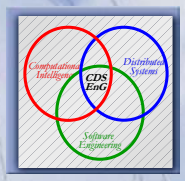
ES595a - Advanced Topics in Software & Systems Design
Cooperative Distributed Systems Engineering: Technologies &Applications



W06: *Agent-Orientation*

Western

Cooperative Distributed Systems (CDS)

- CDS is a class of systems with entities that are able to
  - perform some functions independently and
  - exercise some degree of authority in
  - sharing their capabilities

CDS: Design Concepts^{++} Recall…

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

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

CDS-Design concepts

Autonomy
Coordination
Communication
Interaction
Cooperation
Adaptability

?? Paradigm

Abstraction
Refinement
Modularity
Information Hiding
Interface
Functional Independence
Architecture
Reusability

OO Paradigm

Modeling Interdependency Types–Ghenniwa's

Interdependency types Include:

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

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

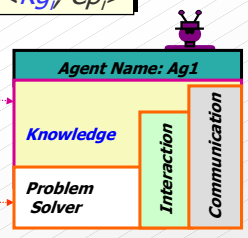
The Agent (System)–Ghenniwa's

- 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
    - The basic components include:
      - knowledge
      - Capabilities
        - problem-solving
        - communication
        - interaction
    - 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 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

CDS-Design concepts

Autonomy
Coordination
Communication
Interaction
Cooperation
Adaptability

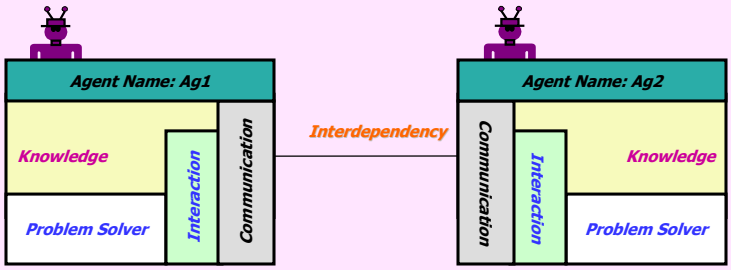
AO Paradigm

Abstraction
Refinement
Modularity
Information Hiding
Interface
Functional Independence
Architecture
Reusability

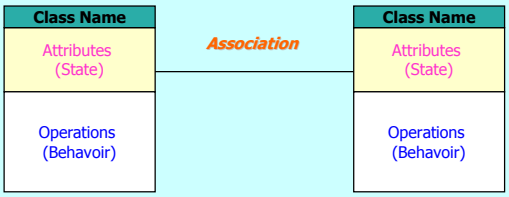
OO Paradigm

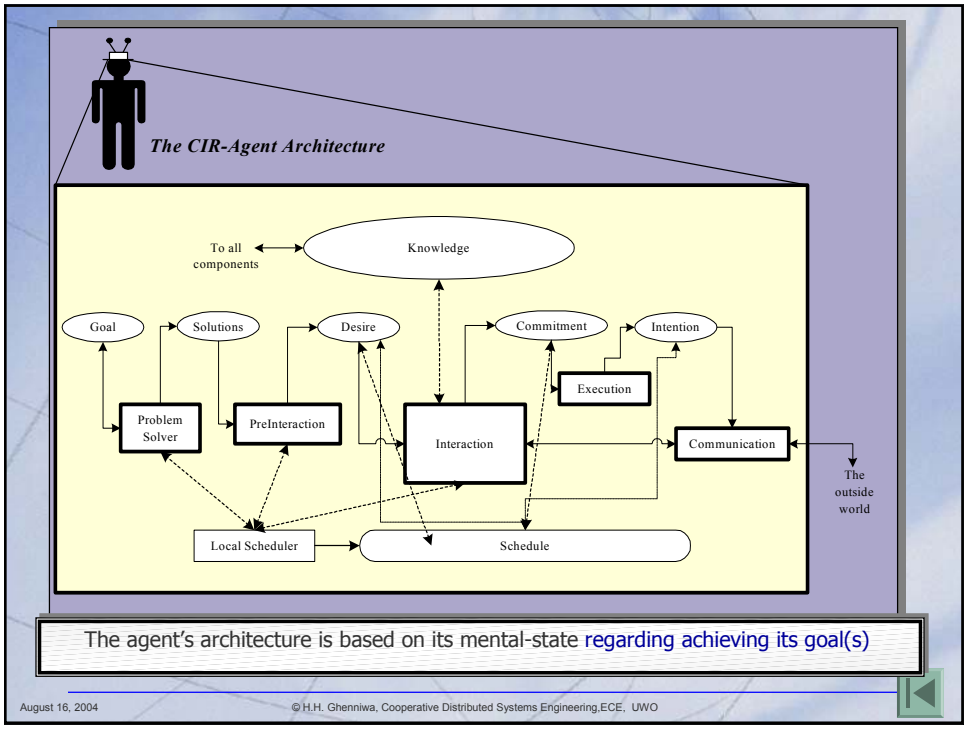
Agent- Logical (Conceptual) Model

AO Paradigm



OO Paradigm





The Coordination Mechanism

- A set of
  - decision points for coordinated control
  - interaction devices

  collectively deal with the interdependency problem

  - Main objectives:
    - reduce and resolve the problems associated with interdependency
  - To deal with the dynamic characteristics of the environment
    - *intelligence* and *rationality* are employed
      - *Intelligence* is a set of heuristics related to the nature of the type of interdependency and the application domain
      - *Rationality* is the ability of the agent to analyze
        - *Why*, *when*, *and which* coordination solution is appropriate according to some performance measure

Reasoning about "Which?" -Coordinated Control

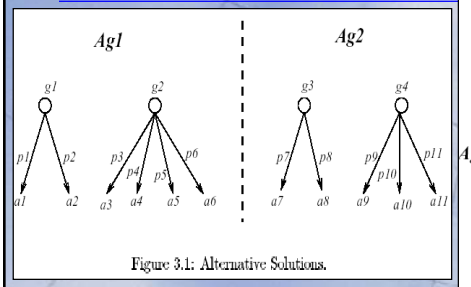


Figure 3.1: Alternative Solutions.

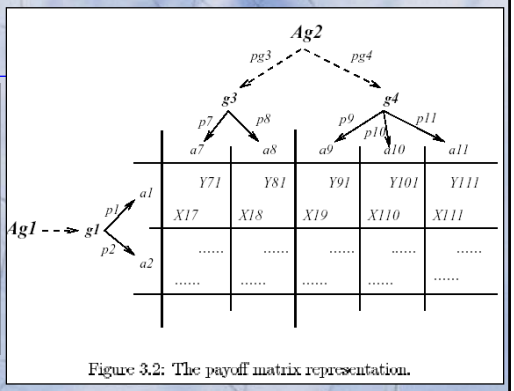


Figure 3.2: The payoff matrix representation.

Interaction Device

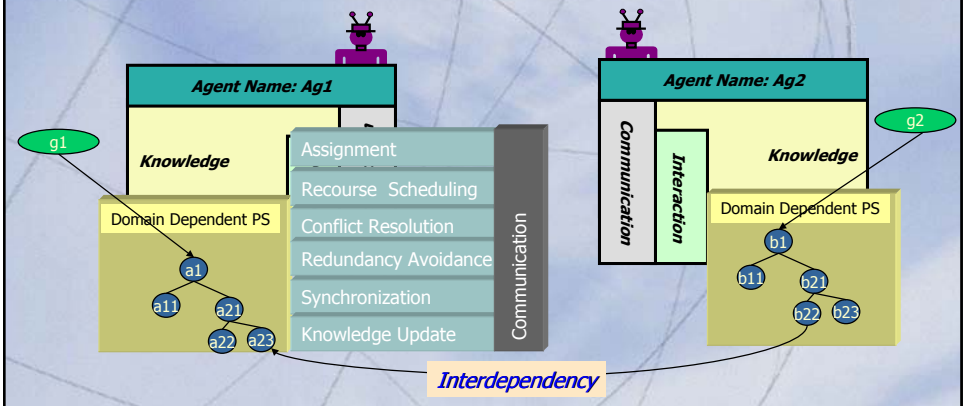
- The basic characteristics of interaction device
  - Problem specifications (PoblemSpec)
  - Evaluation parameters (EvalPara)
  - Sub-processes (Spro)

Assignment
Recourse Scheduling
Conflict Resolution
Redundancy Avoidance
Synchronization
Knowledge Update

IntDev < :DeviceType Dev, :PoblemSpec PSpec, :EvalPara EvPar, :Spro Pro >

Agent Conceptual Model

The CIR-Agent –Autonomy & Rationality @ Coordination



Rationality

- To deal with the dynamic characteristics of the environment
  - rationality is employed at two levels
    - meta-reasoning
    - action-level

Reasoning about "Which?"
-Coordinated Control

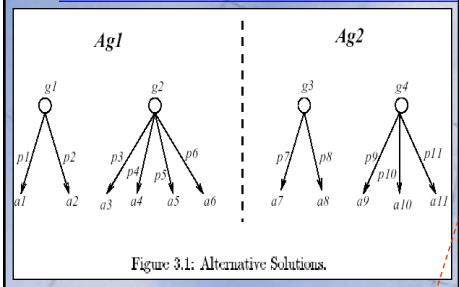


Figure 3.1: Alternative Solutions.

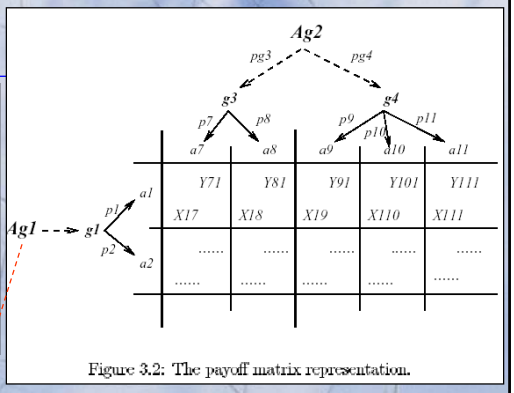


Figure 3.2: The payoff matrix representation.

$$\min_{j\in\{1,2\}} c_1^j = pg_3[p_7X_{j7} + p_8X_{j8}] + pg_4[p_9X_{j9} + p_{10}X_{j10} + p_{11}X_{j11}]$$

$$c_1^1 = pg_3[p_7X_{17} + p_8X_{18}] + pg_4[p_9X_{19} + p_{10}X_{110} + p_{11}X_{111}]$$

$$X_{111} = X_{19} = X_{110} = 0 \qquad X_{18} = X_{17} = X_r$$

$$c_1^1 = pg_3p_7X_r + pg_4p_{10}X_r = [pg_3p_7 + pg_4p_{10}]X_r = p_r^1X_r$$

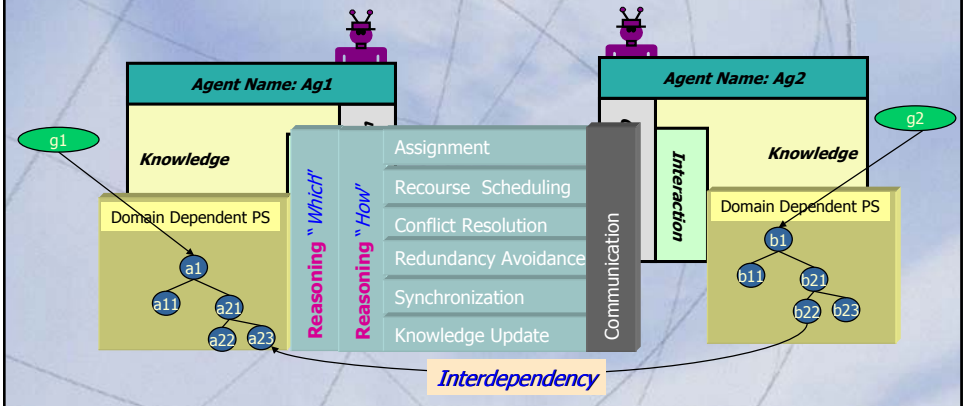
Which interdependency?

Reasoning about "How?"
Interaction Device –*Autonomy & Rationality @ Coordination*

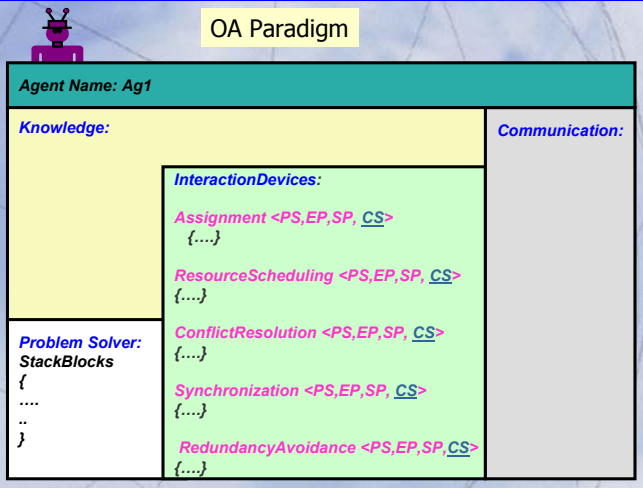
- Characteristics of the Interaction Devices are determined by:
  - The number of the potential participants
  - The upper time-bound and solution-quality
    - The rationality perspective: *Optimal* vs. *Satisficing* Strategies
      - Bounded-optimality
      - Bounded-duration

- Assignment
- Recourse Scheduling
- Conflict Resolution
- Redundancy Avoidance
- Synchronization
- Knowledge Update

Agent Conceptual Model

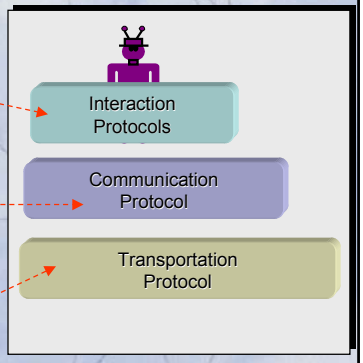


The CIR-Agent

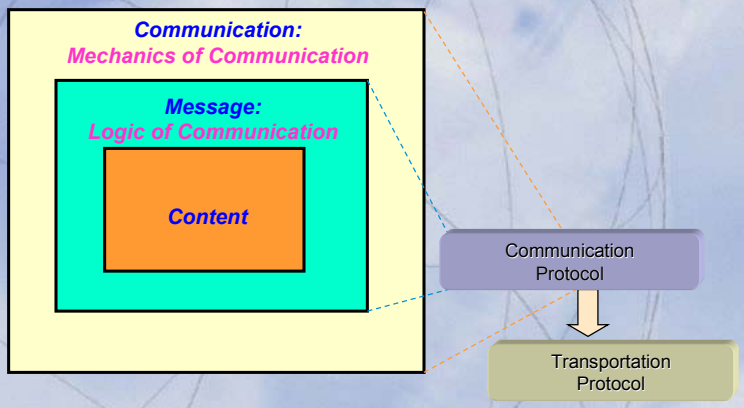


Supporting Levels of Agent Machine

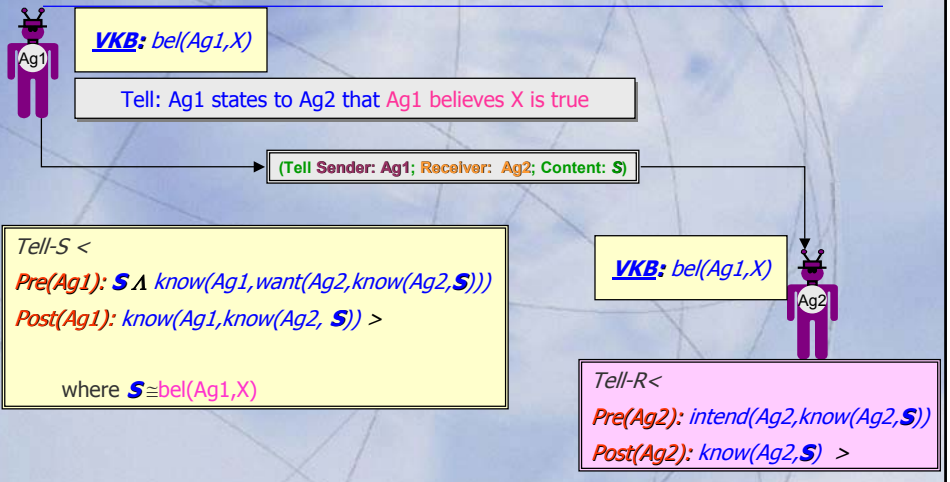
- Interaction layer:
  - Refers to coordination-level strategy pursued by the agents and to the policy managing the structure of the inter-agent conversation
- Communication layer:
  - The medium through which attitudes regarding the content of the exchange are communicated
- Transportation layer:
  - The actual transport mechanism used for the networked communication



ACL is a Layered Language: KQML



Semantics for TELL



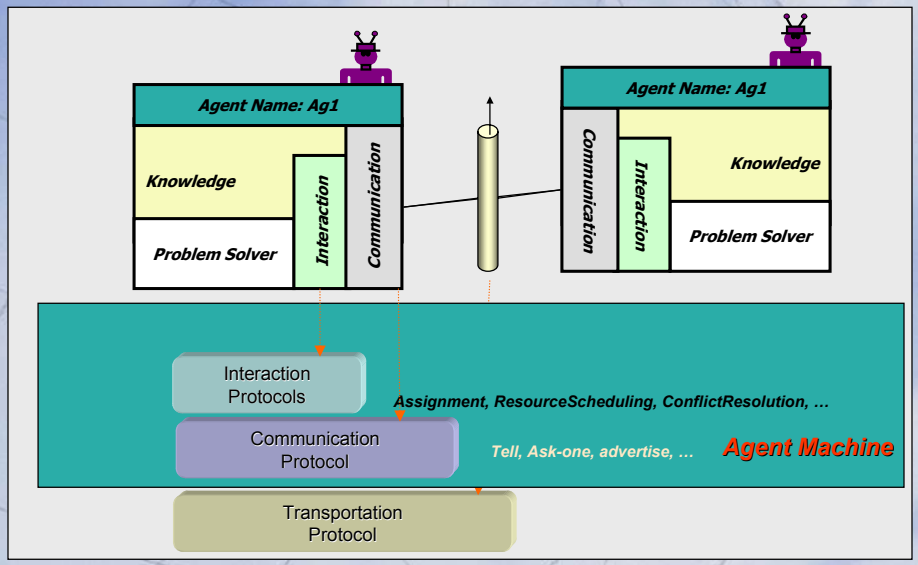
Agent-to-Agent Communication View

Systems ultimately need to communicate up at the semantic levels

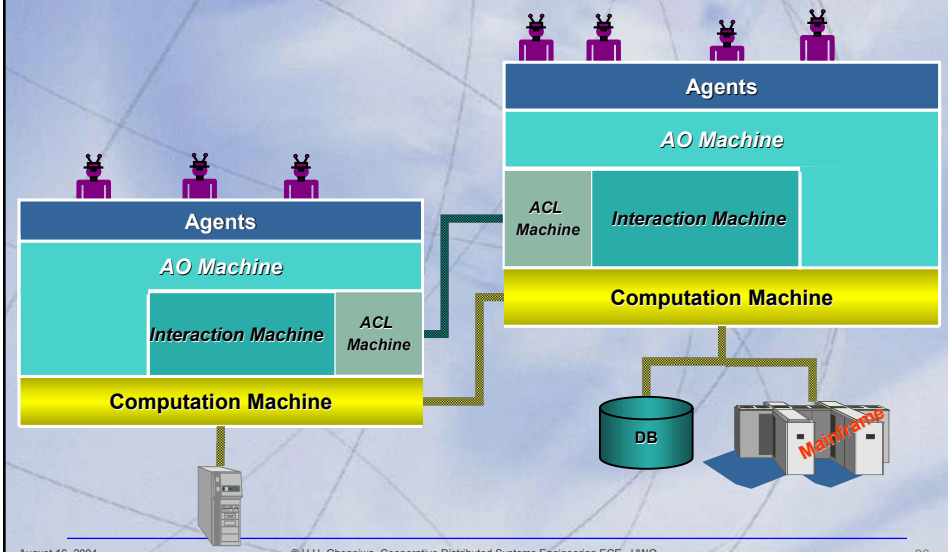
Interaction Protocols	Conversation	⟷	Agent Level (interaction) Protocols
Communication Protocol	Comm. Acts	⟷	Standard ACL's, Sound semantics
	Content	⟷	Logical, XML, RDF
	Ontology	⟷	XML, RDF, BizTalk
Transportation Protocol	Syntax	⟷	HTML
	Protocol	⟷	HTTP
	Transport	⟷	TCP/IP, etc.

Semantic communication based on models of agents

Supporting Levels of Agent Coordination

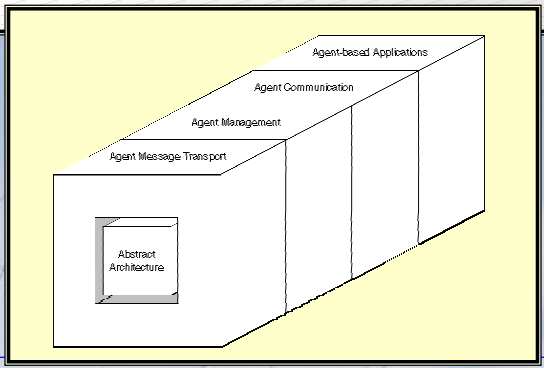


Agent-Oriented Machine

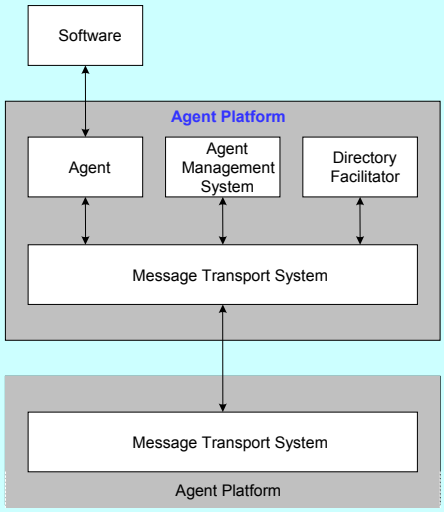


AO Machine: Platform
Example: FIPA

- Foundation for Intelligent Physical Agents
- Structure of standards specifications in the following main areas
  - Management services (AMS)
  - Directory services (DF)
  - Agent communication channel
    - Message transport



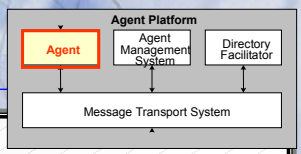
AO Machine: Platform
Example: FIPA



Agent platform (AP)

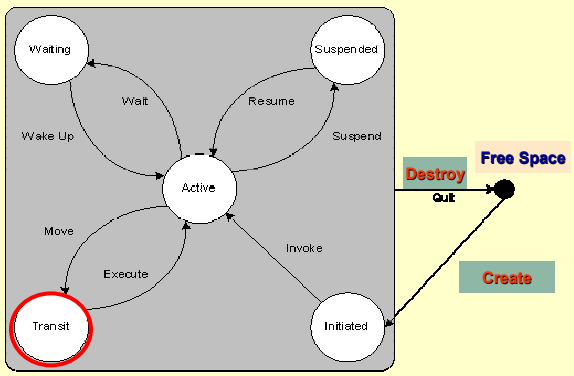
- Agents: Application
- Specialized Agents
  - Directory facilitator
  - Agent management system
- Message transport system
  - Agent communication channel (ACC)
    - The message transport service is the default communication method between agents on different APs
    - ACC supports the message transport system
    - The ACC also supports AP routing tasks

AO Machine: Platform
Example: FIPA Agent



- The agent artifact
  - The fundamental element on an AP
    - Combines one or more functional capabilities into an integrated execution mode
  - May support several notions of identity
    - **Agent identifier** (AID)
      - Labels an agent so that it may be distinguished unambiguously within the agent universe
  - May be registered at a number of transport addresses at which it can be contacted

AO Machine: Platform
Example: FIPA Agent Lifecycle



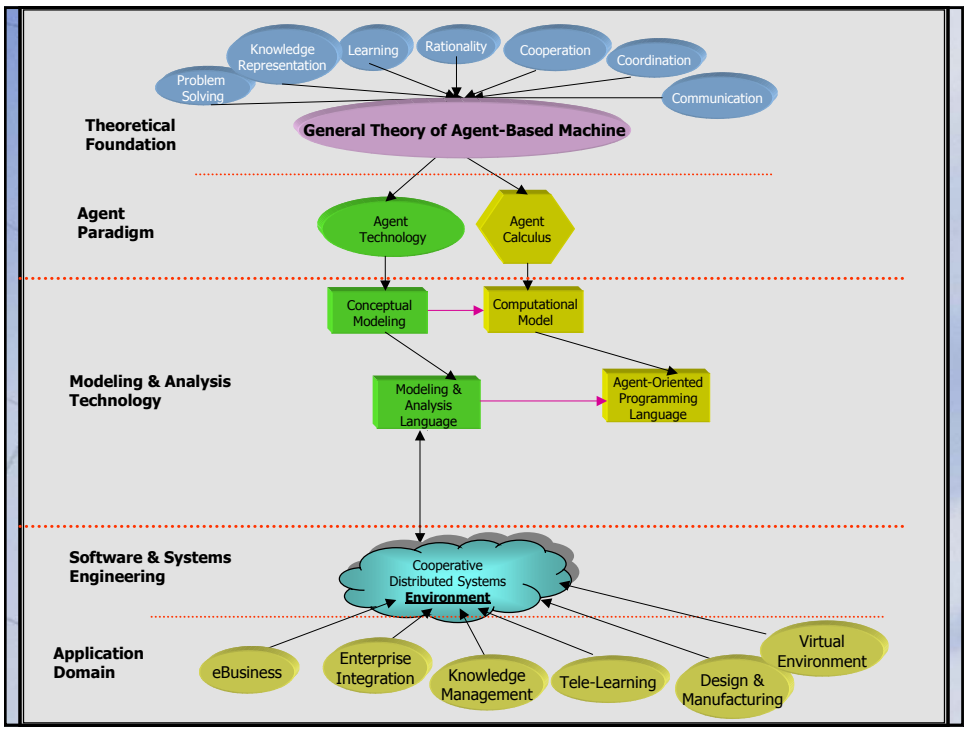
- The lifecycle states can be described in terms of
  - AMS responsibilities to message delivery in each state of the agent's lifecycle

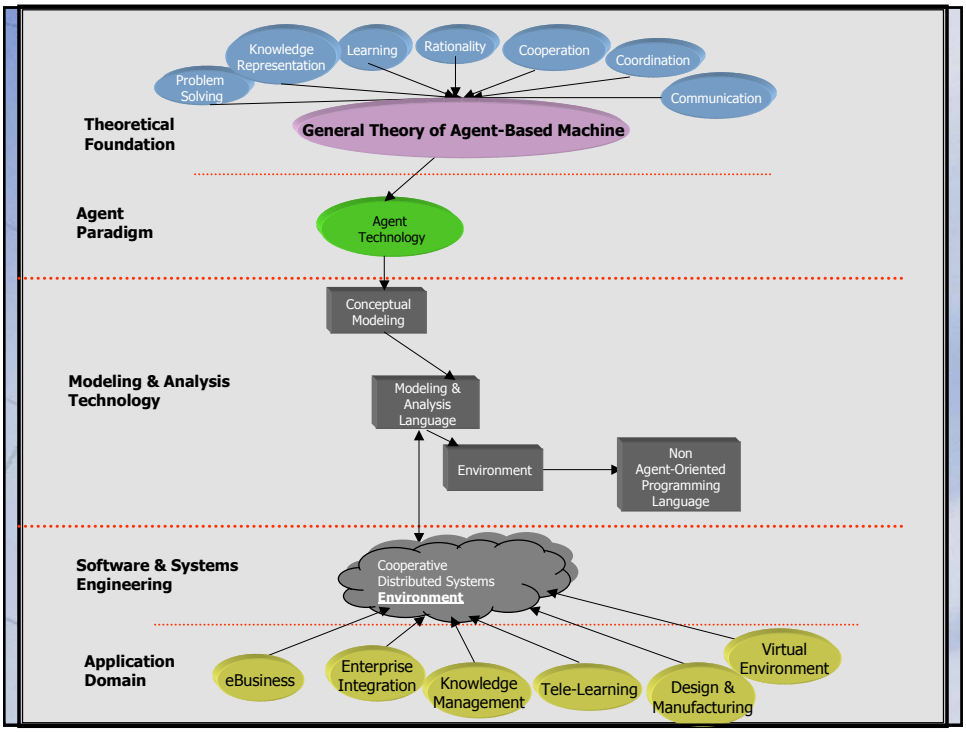
AO Machine:
Transfer of Control & Semantics

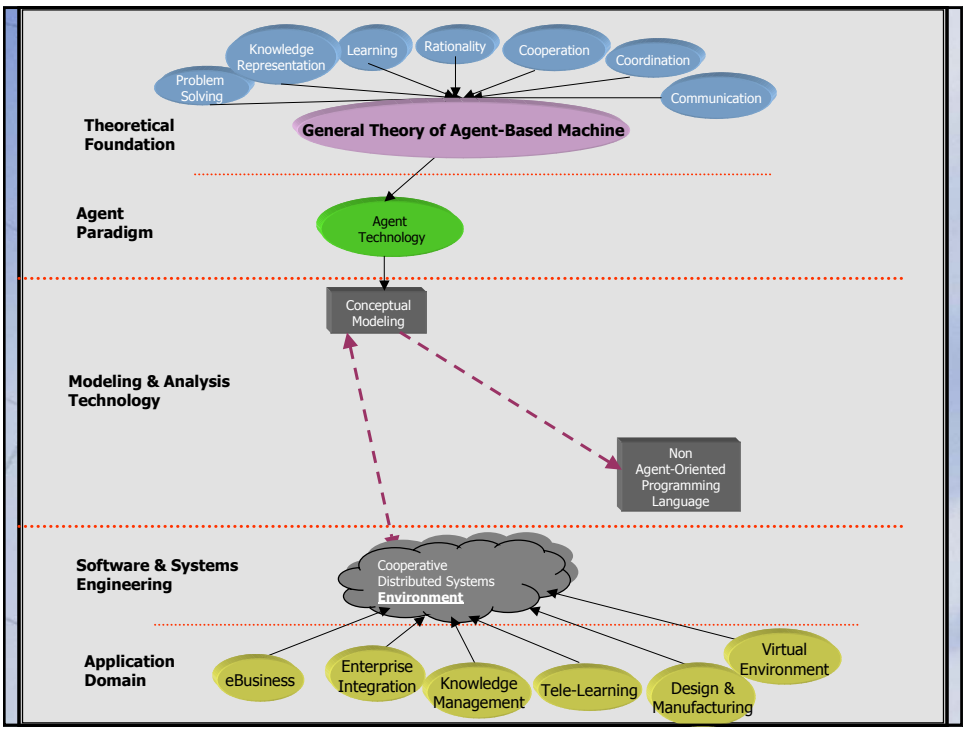
- Transfer of control
  at the coordination level
- Semantics Issues:
  at the communication level

Assignment
Recourse Scheduling
Conflict Resolution
Redundancy Avoidance
Synchronization
Knowledge Update

```
(ask-one
      :content <expression>
      :aspect <expression>
      :language <word>
      :ontology <word>
      :reply-with <expression>
      :sender <word>
      :receiver <word>)
```

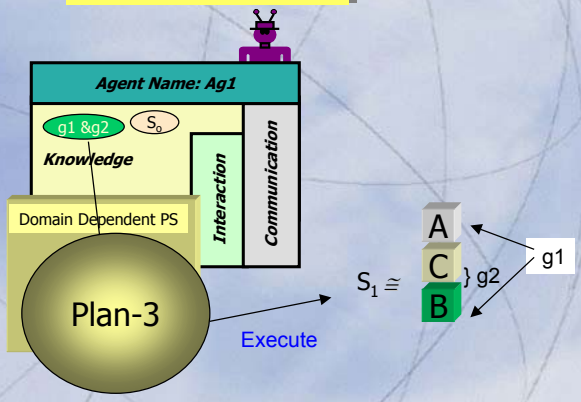

Knowledge Representation
Learning
Rationality
Cooperation
Coordination
Problem Solving
Communication
Theoretical Foundation
General Theory of Agent-Based Machine
Agent Paradigm
Agent Technology
Conceptual Modeling
Modeling & Analysis Technology
Modeling & Analysis Language
Environment
Non Agent-Oriented Programming Language
Software & Systems Engineering
Cooperative Distributed Systems Environment
Application Domain
eBusiness
Enterprise Integration
Knowledge Management
Tele-Learning
Design & Manufacturing
Virtual Environment


Knowledge Representation
Learning
Rationality
Cooperation
Coordination
Problem Solving
Communication
Theoretical Foundation
General Theory of Agent-Based Machine
Agent Paradigm
Agent Technology
Conceptual Modeling
Modeling & Analysis Technology
Non Agent-Oriented Programming Language
Software & Systems Engineering
Cooperative Distributed Systems Environment
Application Domain
eBusiness
Enterprise Integration
Knowledge Management
Tele-Learning
Design & Manufacturing
Virtual Environment

At the application level:
Multiple Goals

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

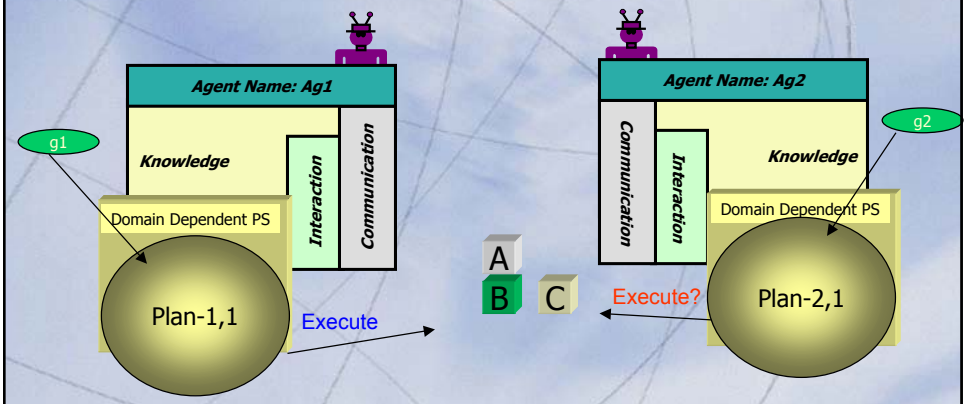
$PS\ (S_0,\ g_1 \& g_2, AC) \cong S1$



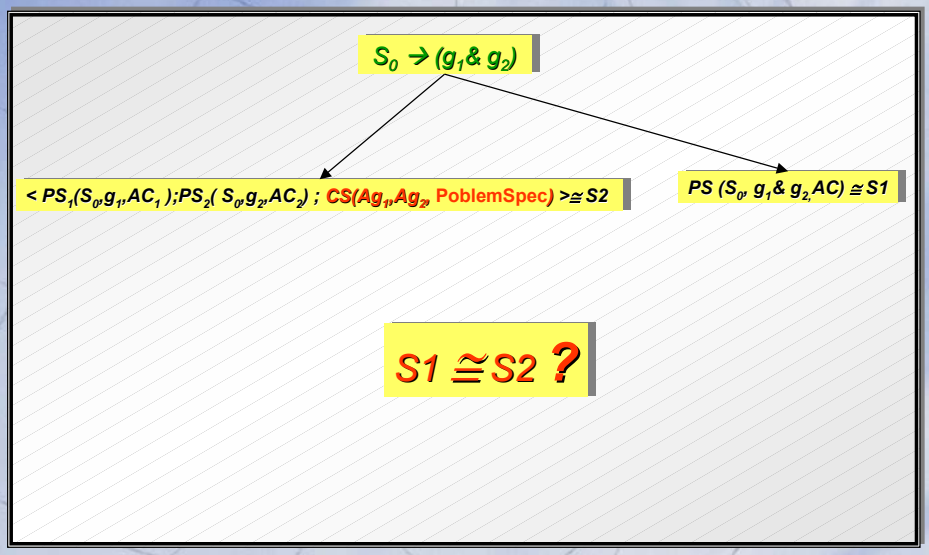
At the application level:
Multiple Agents

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

$< PS_1(S_0, g_1, AC_1);PS_2(S_0, g_2, AC_2)\ ;\ CS(Ag_1, Ag_2,$ PoblemSpec$) > \cong S2$

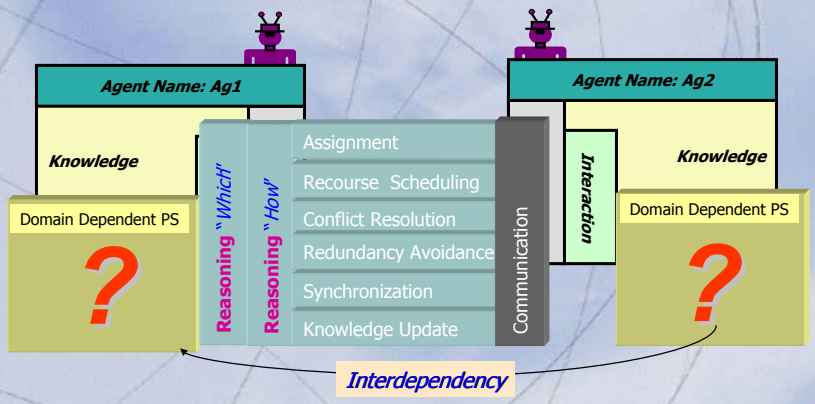


At the application level: Therefore,…

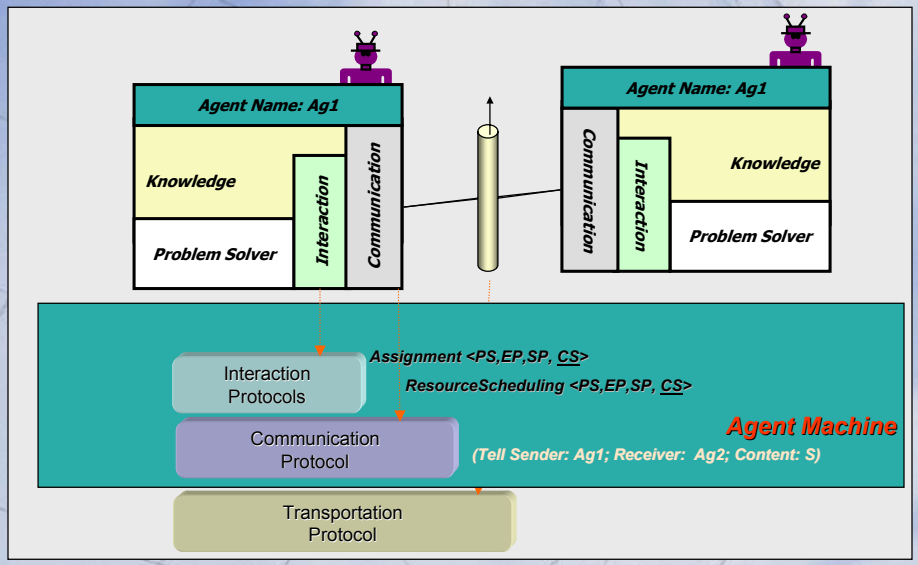


What is missing for CDS?

How About Functionality: PS



What is missing for CDS?



What is Next…

Service-Oriented Computing