

## Multi-Agent System

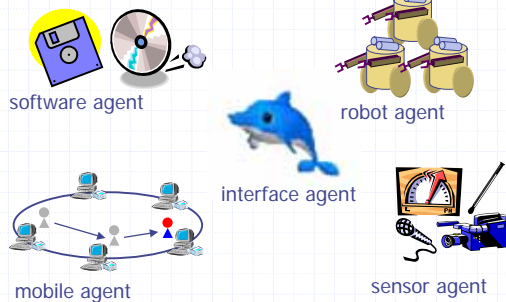
K. Furuta (Univ. of Tokyo)



## What is agent?

- ◆ One that acts or has the power to act
- ◆ Abilities required for an agent
  - Decide one's own behavior autonomously
  - Exchange information with other agents or environment if necessary
  - Respond to unexpected situation to some extent

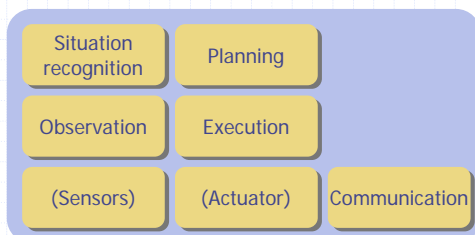
## Various agent types



## Road towards agent

- ◆ FORTRAN (1957)
  - Abstraction of procedure (subroutine)
- ◆ ADA (1975)
  - Modularization & portability (package)
- ◆ Small Talk (1980)
  - Abstraction of data (object)
- ◆ Distributed artificial intelligence (DAI), distributed control, etc ...

## Basic configuration of agent



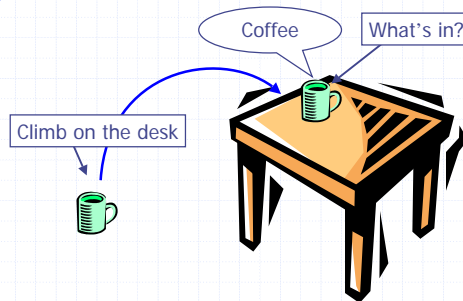
## Supporting technologies

- ◆ Object-oriented programming
  - JAVA, C++, *etc.*
- ◆ Distributed computing
  - CORBA, RMI, HORB, *etc.*
- ◆ Agent communication
  - TCP/IP, IC Card, BlueTooth, ACL, KQML, *etc.*
- ◆ Intelligence or adaptation
  - Inference, planning, pattern recognition, learning, evolution, *etc.*

## Object-oriented programming

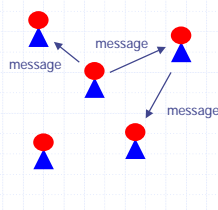
- ◆ Object = unit of software organization
- ◆ Abstracted body of procedure and data
  - Unnecessary to know internal structure
- ◆ Generate instance from class before use
  - Inheritance of variables and methods
- ◆ Call method to handle or communicate with agent

## Image of object-oriented programming



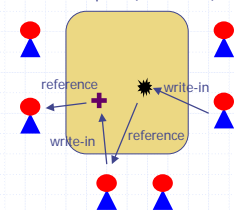
## Agent communication

Direct communication  
with designated partner



Indirect communication

shared space (blackboard)



## Agent Communication Language (ACL)

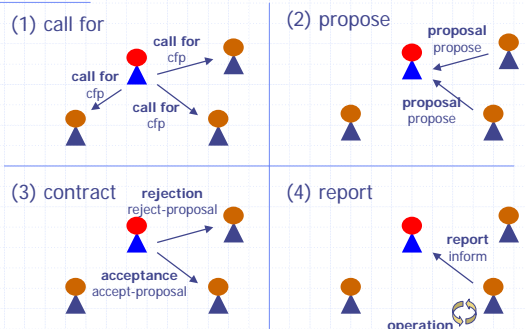
query-if	ask truth-value of a proposition
query-ref	ask object in a proposition
subscribe	request information
inform	general act of notification
inform-if	inform truth-value of a proposition
inform-ref	inform object in a proposition
confirm	confirm verity of a proposition
disconfirm	confirm falseness of a proposition
failure	notify failure of communication
not-understand	notify incomprehensible communication

## Agent Communication Language (ACL)

cfp	call for execution of an action
propose	propose execution of an action
accept-proposal	accept proposal
reject-proposal	reject proposal
request	request execution of an action
request-when	request execution of an action when condition is satisfied
request-whenever	request execution of an condition is satisfied
agree	agree execution of an action
refuse	refuse execution of an action
cancel	cancel execution of an action

## Cooperation of agents

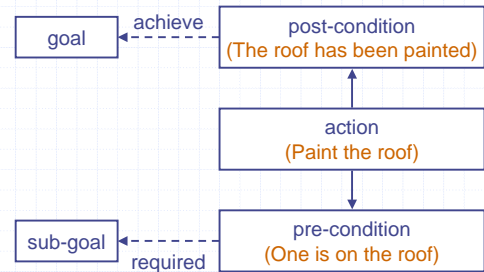
— Contract network protocol —



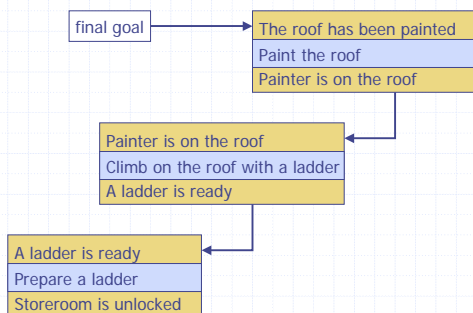
## Planning of agent

- ◆ Elaborative planning
  - Act after plan to the final goal has been completed
  - Efficient if situation does not change
  - Replanning necessary after situational change
- ◆ Responsive planning
  - Execute what can be done in the present situation
  - No guarantee the goal is to be achieved
  - Adaptable to situational change

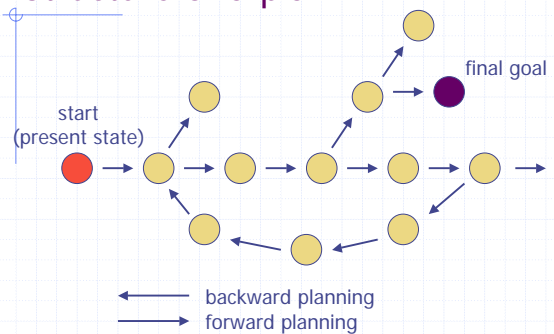
## Knowledge necessary for elaborative planning



## Planning by backward chaining



## Structure of a plan



## Example of responsive planning

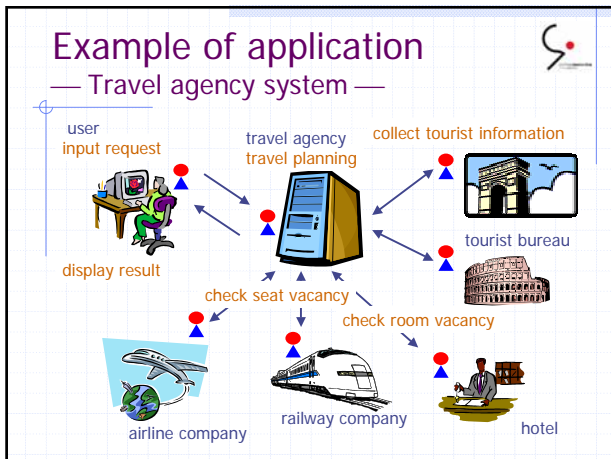
— Subsumption architecture —

A robot that strolls around the room to create a map



## Multi-agent system

- ◆ System that achieves the goal in a self-organizing manner by cooperative behavior of multiple agents
  - Adaptable flexibly to situational change
  - Easy to extend or modify the system
  - Robust against failure of system components



- ### Multi-agent simulation
- ◆ Simulation such that relevant components are modeled based on agent technologies
    - Team simulation
    - Traffic (automobiles, aircrafts) simulation
    - Artificial market
    - Artificial society

