Overview

Agent Architectures

Definition
of agent architecture

“Classical” Architectures for robots
consists of functional components

Situated Automata
eliminates explicit deliberation

Behavior-Based Architectures
behaviors as essential components

Agent Infrastructure
Agencies

Agent Lifecycle
Creation, Registration, Termination
Agent Architecture

Definition

The architecture of an agent defines how the job of generating actions from percepts is organized (adapted from (??Russell:Norvig:95), p. 786)

Variations for different types of agents

- classical vs. behavior-based
- reflex / goal-based
- knowledge-based
- planning
- learning
- ...

this is a widely open field, there is no accepted theory of agent architectures or architecture design
“Classical” Architectures

mainly for autonomous robots

agent design

functional components are used as building blocks,
e.g. perception, learning, planning

functional module

receives specific information from sensors or other modules, processes it, and delivers results to effectors or other modules

world model

centralized, complete

taskable

a goal can be assigned, and a plan to achieve it can be carried out by the agent

learning
explanation-based, mainly via compilation of used plans

Principal drawback: Explicit reasoning about the effects of low-level actions is too expensive to generate real-time behavior
Shakey

an example of a classical robot architecture

vision system
for simple object location

path-planning algorithm
two-dimensional

theorem prover
constructs simple symbolic plans based on the situation calculus

physical components
wheels, motors, sensors, processors

Several improvements for later versions, mainly through special-purpose components (low-level actions, LLAs) and plan compilation (macro-operators)
Situated Automata

concrete specification of simple agents

finite-state machine
  input from sensors, outputs to effectors

reflex agents
  essentially efficient implementations of reflex agents with state

explicit knowledge representation
  generates the automaton by an offline compilation process

decomposition
  manual design process according to various necessary behaviors

Goal: Eliminate explicit deliberation
Problem: Sometimes it is necessary
Behavior-Based Robotics

behaviors as building blocks

agent design
composed from basic behaviors,
e.g. obstacle avoidance, wall-following,
exploration

behavioral module
accesses sensors independently, evaluates
information, sends signals to effectors

prioritized hierarchy
“higher” behaviors can override “lower”
ones

world representation
no need for a centralized, complete
representation
“the world is its own model”

Single, inexpensive mechanism that can achieve
many basic competences in the world.
Problem: A new task requires a complete redesign of the agent
Agencies

Infrastructure for agents

location
   a place where agents can “live”
   not necessarily a physical entity

registration
   keeping track of agents at an agency

services
   an agency often serves as a market place
   for agents that utilize and offer services

control
   resources, agent behavior

expenses
   agencies often provide a mechanism to
   pay and charge for services
Agent Lifecycle

creation
usually initiated by the user
sometimes agents “spawn” other agents

registration with agencies
model for providing and utilizing services

termination
  task is finished
  agent is out of control

task definition
  a new task can be assigned to the agent

mobility
may be viewed as multiple lifecycles at different locations (agencies)
Summary - Agent Architectures

how to organize the generation of actions from percepts

variations for different types of agents
classical, behavior-based, reflex,
goal-based, . . .

classical architectures
functional components as building blocks

situated automata
formal, concise specification of simple agents

behavior-based agents
basic building blocks are behaviors
no centralized world model