Learning Guide Unit 2

Site: University of the People
Course: CS 4408-01 Artificial Intelligence - AY2025-T3
Book: Learning Guide Unit 2
Printed by: Ryohei Hayashi
Date: Thursday, 30 January 2025, 3:43 PM

Description

Learning Guide Unit 2

Table of contents

Overview

Unit 2: Agents

Topics:

- Definitions of agents
- Rational Agent
- Different types of AI Environments
- Agent architectures
- Hierarchical control
- Embedded and simulated agents
- Knowledge based systems
- Nature of agents
    - Autonomous versus semi-autonomous
    - Reflexive, goal-based, and utility-based
    - The importance of perception and environmental interactions

---

Learning Objectives:

By the end of this Unit, you will be able to:

1. Examine the defining characteristics of intelligent agents.
2. Identify the PEAS and environment types of intelligent agents.
3. Create code to implement a rational agent.

Tasks:

- Read the Learning Guide and Reading Assignments
- Participate in the Discussion Assignment (post, comment, and rate in the Discussion Forum)
- Complete and submit the Programming Assignment
- Complete an entry in the Learning Journal
- Take the Self-Quiz

Introduction

In Unit 2, we examine the different architectures of agents including their key characteristics. We further examine the components of how an agent is constructed and how these components work together to carry out the goal or function of the agent. We will also look at the Hierarchical control agent structure which separates the components of the agent into layers each of which has a specific function or purpose.

An agent perceives its environment through sensors. The complete set of inputs at a given time is called a percept. The current percept, or a sequence of percepts, can influence or provide input into the actions of an agent. The agent can change the environment through actuators or effectors. These actuators or effectors may be physical or simulated depending upon the nature of the agent. Embodied agents, for example, have physical actuators that can effect change in the environment of the real world. Software agents, on the other hand, have simulated actuators and can affect change within the virtual environment in which they operate. There are many examples of software agents. One example is the idea of a non-player character (NPC) in a video game. This is a simulated agent that is purposive and is designed to act with reason.

An operation involving an effector is called an action. Actions can be grouped into action sequences. Agents can have goals that they try to achieve. These types of agents are called purposive agents.

Thus, an agent can be looked upon as a system that implements a mapping from

percept sequences to actions. A performance measure has to be used in order to evaluate an agent. An autonomous agent decides autonomously which action to take in the current situation to maximize progress towards its goals. An autonomous agent is one that makes decisions and acts on its own according to the actions it is capable of performing and the rules of reasoning or decision making that it is operating under.

Based on this understanding, we can define an agent using these 4 rules:

1. Ability to Perceive Environment

   The agent should be able to perceive what's happening around it, which means it gets information about the environment.

2. Observations Used to Make Decisions

   An agent should be able to use that perceived information to make a decision.

3. Decisions will Result in Action

   A decision will result in an action being taken from the information that is received.

4. Ability to become Rational

   An AI agent must be rational meaning, whatever actions are taken have to be the best possible action the agent can make based on that perceived information. The rationality of the agent is measured by its performance, prior knowledge, the environment it can perceive, and finally actions it can perform.

Rational Agent

A rational agent is often known as an intelligent agent and the above rules are often grouped in the term as PEAS (Performance, Environment, Actuators, Sensors)

If we consider a self-driving car as an intelligent agent, then it should have the following PEAS:

- Performance: driving capability, safety, comfort, time.
- Environment: road, road signs, other cars, pedestrians, weather.
- Actuators: Steering, brake, accelerator, horn, signal.
- Sensors: camera, sonar, odometer, speedometer, GPS, accelerometer.

Different types of AI Environments

Next, we can classify Environment into the following categories:

- Fully observable and Partially observable:
  If an agent's sensors give access to the complete state of the environment at each point in time, it is known as fully observable, otherwise not or partially observable. For example, chess would be a fully observable environment, while poker is not.

- Deterministic and Stochastic:
  If the "next state" of the environment is determined by the current state and the action executed by an agent then the environment is deterministic. A Stochastic environment is random in nature and cannot be completely determined. For example, the 8-puzzle game has a deterministic environment, but a driverless car is stochastic.

- Static and Dynamic:
  In a static environment, when an agent is deliberating, the environment remains unchanged. An example would be the "Speech analysis" problem in AI, which operates in a static environment. In contrast, dynamic AI environments like "Vision AI systems" in drones deal with data sources that change quite frequently.

- Discrete and Continuous:
  A discrete AI environment consists of a finite set of possibilities (although arbitrarily large) that can drive the final outcome of an action. Chess is classified as a discrete AI problem. On the other hand, continuous AI environments rely on unknown and rapidly changing data sources. Vision systems in drones or self-driving cars would be examples of continuous AI environments.

- Single agent and Multi-agent:
  When an agent is operating by itself, this is known as a single-agent environment. In contrast, if there are other agents involved, then it's a multi-agent environment.

Agent Architecture and Hierarchical Control

An agent is made up of a body and a controller. The body can be physical or virtual. The body is the part of the agent that interacts with the agent's environment. The body has both sensors and actuators.

Sensors are the capabilities of the agent that collect stimuli from the environment. If the agent is a robot then sensors might include things such as GPS which provides the position of the robot within the environment. A compass might be used which provides directional information and informs the agent of the direction that it is facing. A whisker might be used to 'sense' if the robot is about to or has come into contact with another physical object. A camera might be used to provide a visual sense of the environment around the robot and so on. If the agent is a software agent the sensors might be components of software that sample packets from the network, get metrics on how a computer system is operating in terms of memory, CPU, network, or input/output utilization. Perhaps the agent monitors a room to ensure that it remains comfortable to its human inhabitants and which case it may monitor temperature, humidity, or lighting levels. In each of these cases, the sensors are collecting information about the environment in which the agent operates.

Actuators provide the agent with the ability to affect change within the environment that it operates. In the case of a robot, actuators might include motorized wheels that enable the robot to move position or arms that enable the robot to grasp or manipulate other objects within the environment.

For example, the following picture shows robots entered into a student robotics competition where teams of students must design, build, and program a robot capable of carrying out a complex task. In this case, the task is shooting a basketball into the basketball net.

Image credit: North Charleston from North Charleston, SC, United States / CC BY-SA

In addition to a body to interact with its environment, an agent requires a controller. The controller accepts the stimuli sensed by the body in the form of percepts which are data values created from the stimuli that are sensed by the body. The controller is the portion of the agent that must 'reason' how to respond to the stimuli to achieve the goals set for the agent to accomplish. Such reasoning takes the form of actions that the agent much effect within its environment which is done by sending commands to the body directing the actions of its actuators.

Hierarchical control

One approach that can be taken to implement the different components of an agent is the hierarchical control architecture. In this approach, the functionality of the agent is implemented in layers. One layer might represent the body and the interaction with the environment. Another layer might include the logic and reasoning for issuing the commands the agent must execute and still, another might incorporate the logic for developing and following a plan to accomplish the goals of the agent.

Our text explores this form of agent design in detail and one of the assignments for this week will include designing the logic that would be implemented in such a hierarchical agent.

References

TutorialsPoint. (n.d.). *AI-Agents & environments*.
https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_agents_and_environments.htm

Reading Assignment

Poole, D. L., & Mackworth, A. K. (2017). *Artificial Intelligence: Foundations of computational agents.* Cambridge University Press. https://artint.info/2e/html/ArtInt2e.html

Read the following chapters:

- Chapter 2 – Agent Architectures and Hierarchical Control

Video Resources

Taipala, D. (2014, September 12). *CS 4408 artificial intelligence unit 2 lecture 1* [Video]. YouTube.

Taipala, D. (2014, September 12). *CS 4408 artificial intelligence unit 2 lecture 2* [Video]. YouTube.

Discussion Assignment

For each of the following agents, describe all the possible PEAS (Performance, Environment, Actuators, and Sensors) and then identify the environment type.

- Self-vacuum cleaner
- Speech recognition device
- Drone searching for a missing person
- Computer playing chess
- Alexa/Siri performing as a smart personal assistant.

Your Discussion should be at least 250 words in length, but not more than 750 words. Use APA citations and references for the textbook and any other sources used.

Programming Assignment

In this assignment, you will be designing a Rational Agent which cleans a room with minimum possible actions and the goal is the entire environment is clean and agent back to home (starting location A)

Part A

In Part A, you need to identify the PEAS (Performance Measure, Environment, Actuators, and Sensors)

Part B

You are required to write pseudo-code in Java or Python to implement this Rational Agent

Your pseudo-code must represent the following:

1. Implementation of the static environment. You need to define array or some other data structure which will represent the location (A to P)
2. A function/method to determine what action to take. Decision should be Move (Which direction), suck dirt, or Go back Home.
3. A function/method to determine which direction to go
4. A function/method to identify the route and navigate to home from the current location.
5. A function/method to test if the desired goal is achieved or not.

Part C

Here, you are required to write 1-page minimum summary focusing on the following thoughts:

- Your lessons learned from doing this assignment
- Do you think your algorithm should be able to reach the goal with given energy points (100)?
- What changes do you need to make if your static location gets bigger?
- The real world is absurdly complex, for a smart home vacuum cleaner need to consider obstacles like furniture, human or pet. What changes you need to make to handle these obstacles.

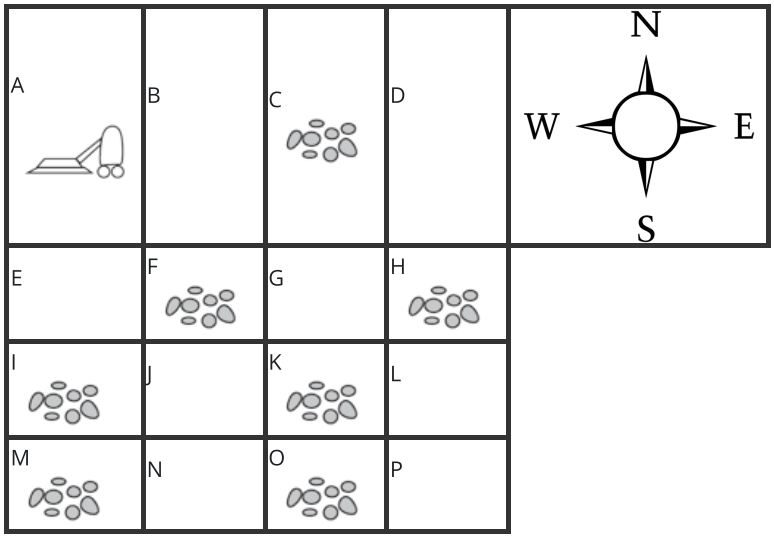

Fig 1: A vacuum-cleaner world with 16 locations (A to P); some locations with dirt and some are not.

Percept Sequence	Action

Percept Sequence	Action
[A, Clean]	Move East
[A, Dirty]	Suck
[D, Dirty]	Suck
[D, Clean]	Move South

Fig 2: Partial tabulation of a simple agent function for the vacuum-cleaner world problem.

Prior Knowledge:

1. The entire environment is divided by 4 by 4 square location.
2. The agent (vacuum cleaner) has an initial energy of 100 points.
3. The agent can move only North, South, East, or West. It can't move diagonally.
4. Each action cost 1 energy point. For example, each move cost 1 energy, each suck cost 1 energy.
5. The agent has a bag that collects dirt. The maximum capacity is 10
6. After each suck, the agent needs to check its bag, if full then go back to Home (location A) and self-empty the bag and start vacuuming again.

You will be graded on the following:

1. Part A, the PEAS is defined correctly for a Vacuum cleaner agent? (20 points)
2. Part B, the data structure is appropriate for such an environment? (10 points)
3. Part B, the pseudo-code is clearly defined for what action to take in different scenarios? (30 points)
4. Part B, the pseudo-code is clearly defined for whether the agent knows a goal state is achieved or not? (10 points)
5. Part B, the calculation for energy and bag capacity (full yet?) is done correctly? For example, does the agent know when to head back home for self-empty the bag? (10 points)
6. Part C, all of the questions clearly answered in a minimum 1-page summary? (20 points)

Learning Journal

The Learning Journal is a tool for self-reflection on the learning process. The Learning Journal will be assessed by your instructor as part of your Final Grade.

Your learning journal entry must be a reflective statement that considers the following questions:

1. Describe what you did. This does not mean that you copy and paste from what you have posted or the assignments you have prepared. You need to describe what you did and how you did it.

2. Describe your reactions to what you did.

3. Describe any feedback you received or any specific interactions you had while participating discussion forum or the assignment Discuss how they were helpful.

4. Describe your feelings and attitudes.

5. Describe what you learned. You can think of one or more topics and explain your understanding in writings. Feel free to add any diagram or coding example if that helps you explain better.

6. Did you face any challenges while doing the discussion or the development assignment? Were you able to solve it by yourself?

The Learning Journal entry should be a minimum of 400 words and not more than 750 words. Use APA citations and references if you use ideas from the readings or other sources.

Self-Quiz

The Self-Quiz gives you an opportunity to self-assess your knowledge of what you have learned so far.

The results of the Self-Quiz do not count towards your final grade, but the quiz is an important part of the University's learning process and it is expected that you will take it to ensure understanding of the materials presented. Reviewing and analyzing your results will help you perform better on future Graded Quizzes and the Final Exam.

Please access the Self-Quiz on the main course homepage; it will be listed inside the Unit.

Checklist

- Read the Learning Guide and Reading Assignments
- Participate in the Discussion Assignment (post, comment, and rate in the Discussion Forum)
- Complete and submit the Programming Assignment
- Complete an entry in the Learning Journal
- Take the Self-Quiz