SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
Data Science Department
Program: BTech Integrated Sem VI

Course: Application of Artificial Intelligence and Agentic Systems
2026-2027

PART A

(Part A: TO BE REFERRED BY STUDENTS)

Experiment No. 1

Roll No.E004	Name: Rohan Dedhia
Program :BTI	Division:E
Batch:E1	Date of Experiment: 21-07-2026
Date of Submission:	Grade :

A.1 Aim: To be an intelligent agent (vacuum cleaner) using simplex reflex and model-based agent

A.2 Prerequisite:

Basic knowledge about Python functions

A.3 Simple Reflex Agent Code:

[1]:
```
print("=" * 50)
print("SIMPLE REFLEX AGENT")
print("=" * 50)no
while True:
    location = input("Enter location (A/B): ").upper()
    status = input("Enter status (Dirty/Clean): ").capitalize()
    if status == "Dirty":
        print("Action: Suck")
    elif location == "A":
        print("Action: Move Right")
    elif location == "B":
        print("Action: Move Left")
    else:
        print("Invalid Input")
    choice = input("Continue? (yes/no): ").lower()
    if choice != "yes":
        break
```

```
==================================================
SIMPLE REFLEX AGENT
==================================================
Enter location (A/B):  B
Enter status (Dirty/Clean):  Dirty
Action: Suck
Continue? (yes/no):  yes
Enter location (A/B):  B
Enter status (Dirty/Clean):  Clean
Action: Move Left
Continue? (yes/no):  yes
Enter location (A/B):  A
Enter status (Dirty/Clean):  Dirty
Action: Suck
Continue? (yes/no):  no
```

Inference

The **Simple Reflex Agent** decides its action using only the current room and its condition. It does not store or recall any previous information. If the current room is **Dirty**, the agent performs the **Suck** action; otherwise, it moves to the opposite room. The process continues until the user chooses to exit the program.

Analysis

- Makes decisions based only on the current perception.
- Does not maintain any memory of previously visited rooms.
- Cannot recognize whether all rooms have already been cleaned.
- May continue moving between rooms even when both are clean.
- Best suited for simple and fully observable environments where only the present state is considered.

A.3 Model-Based Agent Code:

[2]:
```
environment = {
    "A": "Dirty",
    "B": "Dirty"
}
def model_based_agent(location):
    """
    Model-Based Reflex Agent
    Uses memory of environment.
    """
    global environment
    if environment[location] == "Dirty":
        environment[location] = "Clean"
        return "Suck"
    elif location == "A":
        return "Move Right"
    else:
        return "Move Left"
current = "A"
print("\nInitial Environment")
print(environment)
print("\nExecution Starts\n")
print("{:<6} {:<12} {:<12} {:<15} {}".format(
    "Step",
    "Current",
    "Status",
    "Action",
    "Memory"
))
print("-" * 70)
for step in range(1, 7):
    status = environment[current]
    action = model_based_agent(current)
    print("{:<6} {:<12} {:<12} {:<15} {}".format(
        step,
        current,
        status,
```

```
print("-" * 70)
for step in range(1, 7):
    status = environment[current]
    action = model_based_agent(current)
    print("{:<6} {:<12} {:<12} {:<15} {}".format(
        step,
        current,
        status,
        action,
        environment.copy()
    ))
    if action == "Move Right":
        current = "B"
    elif action == "Move Left":
        current = "A"
print("\nFinal Environment")
print(environment)
print("\nAll rooms are clean.")
```

```
Initial Environment
{'A': 'Dirty', 'B': 'Dirty'}

Execution Starts

Step   Current      Status       Action          Memory
----------------------------------------------------------------------
1      A            Dirty        Suck            {'A': 'Clean', 'B': 'Dirty'}
2      A            Clean        Move Right      {'A': 'Clean', 'B': 'Dirty'}
3      B            Dirty        Suck            {'A': 'Clean', 'B': 'Clean'}
4      B            Clean        Move Left       {'A': 'Clean', 'B': 'Clean'}
5      A            Clean        Move Right      {'A': 'Clean', 'B': 'Clean'}
6      B            Clean        Move Left       {'A': 'Clean', 'B': 'Clean'}

Final Environment
{'A': 'Clean', 'B': 'Clean'}

All rooms are clean.
```

Inference

The Model-Based Reflex Agent maintains an **internal model (memory)** of the environment. It updates the status of each room after every observation. If a room is dirty, it cleans it and updates the memory. When the memory indicates that both Room A and Room B are clean, the agent stops automatically.

Analysis

- Maintains an **internal memory (model)** of Room A and Room B.
- Updates the model whenever new information is received.
- Makes decisions using both the **current percept** and **stored knowledge**.
- Stops automatically when the environment is completely clean.
- More intelligent and efficient than the Simple Reflex Agent because it avoids unnecessary actions.

Complete Execution Table

Step	Current Room	Room Status	Action	Memory After Action
1	A	Dirty	Suck	A=Clean, B=Dirty
2	A	Clean	Move Right	A=Clean, B=Dirty

3	B	Dirty	Suck	A=Clean, B=Clean
4	B	Clean	Move Left	A=Clean, B=Clean
5	A	Clean	Move Right	A=Clean, B=Clean
6	B	Clean	Move Left	A=Clean, B=Clean

Difference Between the Two Agents

Simple Reflex Agent	Model-Based Reflex Agent
No memory	Has memory
Uses only current input	Uses current input + stored state
Cannot remember previous actions	Remembers previous actions
Suitable for fully observable environments	Suitable for partially observable environments
Simpler to implement	More intelligent and flexible

- **Simple Reflex Agent:** Imagine a person who enters a room, looks around, and cleans it if it is dirty. After leaving, they forget everything. If they come back later, they have no memory of whether they cleaned it before.
- **Model-Based Reflex Agent:** Imagine a person carrying a notebook. After cleaning Room A, they write "Room A – Clean." When they return, they check the notebook before deciding what to do. The notebook is the agent's **internal model (memory)**.

This analogy clearly illustrates why a model-based reflex agent makes more informed decisions than a simple reflex agent.