**Introduction**

Agent - Aunonomous entity that is able to solve problems and work in an open and dynamic environment

Autonomous - able to act without the intervention of humans or other systems: they have control both over their internal state and over their behaviour

The agent usually has a limited and partial perception of its environment.

The agent has a partial control over its environment. This means that the same action performed twice in apparently identical circumstances might appear to have entirely different effects and it may fail.

A close-up of a list of words

AI-generated content may be incorrect.

A white rectangular grid with black text

AI-generated content may be incorrect.

**Reactiveness** - A reactive system is one that maintains an ongoing interaction with its environment and responds to changes that occur in it (in time for the response to be useful).

**Proactiveness** - Pro-activeness = generating and attempting to achieve goals; not driven solely by events; taking the initiative.

**Social ability** - is the ability to interact with other agents (and possibly humans) via some kind of agent- communication language and perhaps cooperate with others.

**Learning and adaptation -** refers to an autonomous agent’s ability to improve its performance over time by acquiring new knowledge, refining its behavior, and adjusting to changes in its environment.

**Veracity:** an agent will not knowingly communicate false information.

**Benevolence:** agents do not have conflicting goals, and that every agent will therefore always try to do what is asked for

**Rationality:** agent will act to achieve its goals, and will not act in such a way as to prevent its goals being achieved -at least insofar as its beliefs permit

A diagram of a diagram

AI-generated content may be incorrect.

**Accessible environment** - is one where agents can obtain complete, accurate, up-to-date information about the environment’s state. Most moderately complex environments (including the physical world) are inaccessible. The more accessible an environment is, the simpler is to build an agent to operate in it.

**Deterministic environment** - is one in which an action has a single guaranteed effect -there is no uncertainty about the state that will result from performing an action. The physical world can be regarded as non-deterministic. Non-deterministic environments present greater problems to agent designers.

**Static environment -** can be assumed to remain unchanged except by the performance of actions by the agent (example, blocks world). A dynamic environment is one that has other processes operating on it, and which hence changes in ways beyond the agent’s control. The physical world is highly dynamic.

**Discrete environment** - if there are a fixed, finite number of actions and percepts in it. Russell and give a chess game as an example of a discrete environment and taxi driving as an example of a continuous one.

The **ideal environment** (from an engineer perspective):

* Accessible
* Deterministic
* Static
* Discrete.

**Real world:**

* Inaccessible
* Non-deterministic
* Dynamic
* Continuous.

Autonomy and Decentralization

* Agents are self-contained and make decisions without a central controller.
* Scalability: Unlike monolithic architectures, MAS scales naturally as new agents can be added without modifying existing ones.
* Fault Tolerance: Since control is distributed, the system remains operational even if some
* agents fail.

Interaction and Collaboration

* Agents communicate using high-level protocols (e.g., FIPA ACL). Dynamic
* Adaptation: Agents can change their behavior based on interactions with other agents and the environment.
* Emergent Behavior: MAS allows complex system behaviors to arise from simple agent interactions.

Proactive and Goal-Oriented Design

* Unlike procedural or object-oriented systems, MAS components (agents) are goal-driven.
* Decision-making is local to agents, meaning they can pursue their objectives autonomously while adapting to dynamic environments.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A black and white text on a white background

AI-generated content may be incorrect.A screenshot of a computer code

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Context switching

* Multithreading context switching is preemptive → The OS forces switches, adding overhead.
* Asyncio context switching is cooperative → Tasks switch only when they await, making it more efficient for I/O-heavy applications.

Coroutines are functions that can be paused and resumed, allowing other code to run in the meantime.

**Future in asyncio** is a placeholder for a result that is not yet available. It represents an operation that will complete in the future, but we don't know when.

**Await** suspends execution of the current coroutine and waits for an awaitable

**Task** is a wrapper around a coroutine that allows it to run concurrently in the event loop

A white sheet with black text

AI-generated content may be incorrect.

A white paper with blue text

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a text

AI-generated content may be incorrect.A white paper with black text

AI-generated content may be incorrect.A white background with black text

AI-generated content may be incorrect.A white paper with black text

AI-generated content may be incorrect.







