SISTEMAS MULTIAGENTES

ING. OTTO PARRA GONZÁLEZ, PhD

Chapter 2: Multiagent systems

SISTEMAS MULTIAGENTES


collaborative
design
civil
agent
interactive
environments
elderly
methods
complex
disagreement
multiagent
library
agent-based
infrastructure
simulations
open-source
partitioning
based
intelligent
agents
convex
behaviors
objects
semantic
draw
systems
editors
health
social
integrating
managed
support
animate
making
animation
problems
learning
jason
socially
approach
oriented
cheating
j-madem
groups
projects
java
performance
operational
web
strategic
achieving
simulation
decision
skills
characterization
market-based
platform
asynchronous
simulating
crowd
mas
model
review
peer
system
multi-agent
towards
people
semantics
mechanism
framework
virtual
rational
control
distributed
mobility
navigation
urban
planning
simulator

Topics

2. SISTEMAS MULTIAGENTES				
1. Introducción 2. Ideas generales de los sistemas multiagentes 3. La interacción en los Sistemas Multiagentes 4. Comunicación 5. Coordinación	4	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Exposición de material docente Prueba sobre el capítulo	8 horas 2 horas
		APRENDIZAJE AUTÓNOMO (AA)	Investigación de KIF y KQML	4 horas

INTRODUCTION

- The body of available information is rapidly increasing every day. It is becoming impossible for humans to peruse all the available information and extract the data related to a specific topic. A great deal of important information is being neglected primarily for two major reasons.
  - Firstly, highly relevant information is losing its true meaning within a pool of insignificant and unauthorized information. A lot of available information is published which is of no significance and sometimes may even be destructive and corruptive in nature.
  - Secondly, new knowledge and discoveries are emerging quickly as we continue to progress and new technologies emerge. It is becoming more and more difficult each day to find specific information even within a body of authorized information.
- As a solution to this problem, humans are designing software agents that will do the search for them.

INTRODUCTION

- Social, political, and technological pressure towards intelligent systems able to help and support humans in any non-trivial process and activity is leading the way for new programming paradigms, providing suitable abstractions and mechanisms for modelling and designing complex software systems.
- More the twenty years of academic research on multi-agent systems (MAS) have promoted agent-oriented models and technologies as the most suitable candidates for the design and development of distributed and intelligent applications in complex and dynamic environments.

INTRODUCTION

- Multiagent systems (MAS) are computer systems that consist of multiple agents that interact with each other to achieve their individual and/or collective goals.
- These agents can be software programs, robots, or even human beings, and they are designed to work together in a coordinated manner to solve complex problems or perform tasks.
- The architecture of a multiagent system defines the structure and organization of these agents and how they communicate and cooperate to achieve their objectives.

INTRODUCTION

- Multi-agent systems (MAS) is an expanding field in science and engineering. It merges classical fields like game theory (gamification) and decentralized control with modern ones like machine learning and computer science.
- MAS spans disciplines as diverse as biology, ecology, computer simulation, business, economic science, policy, social sciences, political science, military studies and many others.

INTRODUCTION

- During the last two decades, the idea of Semantic Web has received a great deal of attention. An extensive body of knowledge has emerged to describe technologies that seek to help us create and use aspects of the Semantic Web. Ontology and agent-based technologies are understood to be the two important technologies here.
- A large number of articles and a number of books exist to describe the use individually of the two technologies and the design of systems that use each of these technologies individually, but little focus has been given on how one can design systems that carryout integrated use of the two different technologies.

AGENT

CHAPTER 2

AGENT: DEFINITION

- As Russell and Norvig stated an agent is anything that can be considered able to perceive its environment through sensors and act on this environment through actuators.
- To Macal, an agent shall have the following characteristics:
  - (i) be identifiable—a discrete individual with a set of features and rules (mathematical or logic) that govern behaviour and decision-making capacity;
  - (ii) be located—settled in an environment with which it interacts and also in which interacts with other agents;
  - (iii) be goal-driven;
  - (iv) be self-contained; and
  - (v) be flexible, and have the ability to learn and adapt its behaviour through time-based experiences.

AGENTS

- Most authors agree that although there are multiple definitions of the term ‘agent’, several attributes can be pointed out such as heterogeneity, autonomy, capacity to process and exchange information, follow if-then rules, goal-driven, and deductive code-based units, with boundary and state.
- There are also core behaviours: mobility, interaction, adaptation, and bounded rationality.

AGENTS

- An agent may be goal-driven and takes independent actions to reach its goals. Thus, agents compare behaviour outcomes to its goals and adapt responses in the future. An agent’s behaviour can be described by simple if-then rules that used to describe the theoretical assumptions of agent behaviour, in the form of computational procedures that lead to goal achievement.
- These procedures constitute a plan for achieving agents’ objectives.

AGENT-BASED SOFTWARE

CHAPTER 2

AGENT-BASED SOFTWARE (ABS)

- ABS is one where an agent is used as key abstraction.
- Theoretically, an ABS could be conceptualized in terms of agents and still be implemented without using any software consistent to agents. There is an obvious parallelism with object-oriented software, where it is fully conceivable to design a system based on objects, and implement it without getting use of object-oriented software. Nevertheless, this would be counterproductive or, at least, unusual. The same happens with ABS, where users expect agents to be designed and implemented using agent paradigm (e.g., using specific agent based software).

AGENT-BASED SOFTWARE (ABS)

- One should note that an ABS may have any non-zero amount of agents.
- A multi-agent system (MAS), designed and implemented by means of several interacting agents, is more general and pointedly more complex than the unitary (single case) agent.
- In real world, there are various number of situations where the single-agent case is suitable. A good example is the expert assistant, where an agent acts like an expert assistant to a user attempting to fulfil some task on a computer.

AGENT-BASED SOFTWARE (ABS)

- MAS is a computer-based environment made of multiple interacting intelligent agents. MAS are preferably used in solving problems that are difficult (or impossible) for an individual agent.
- MAS is defined as ‘a loosely coupled network of problem-solving entities (agents) that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity (agent)’.

AGENT-BASED MODEL (ABM)

CHAPTER 2

AGENT-BASED MODEL (ABM)

- As in MAS, agent-based model (ABM) also consists of interacting agents within a specific environment.
- In a computer science (e.g. AI), ABM usually states a computer-based method for studying the (inter)actions of a set of autonomous entities. In non-computing–related scientific domains as in social sciences, ABM could refer to an actor in the social world and be called agent based social simulation (ABSS).

AGENT-BASED MODEL (ABM)

- Davidsson using different combinations of focus areas (e.g. agent-based computing, computer simulation, and social sciences), further subdivides ABSS into three categories:
  - (i) social aspects of agent systems (SAAS);
  - (ii) multi-agent based simulation (MABS); and
  - (iii) social simulation (SocSim).

AGENT-BASED MODEL (ABM)

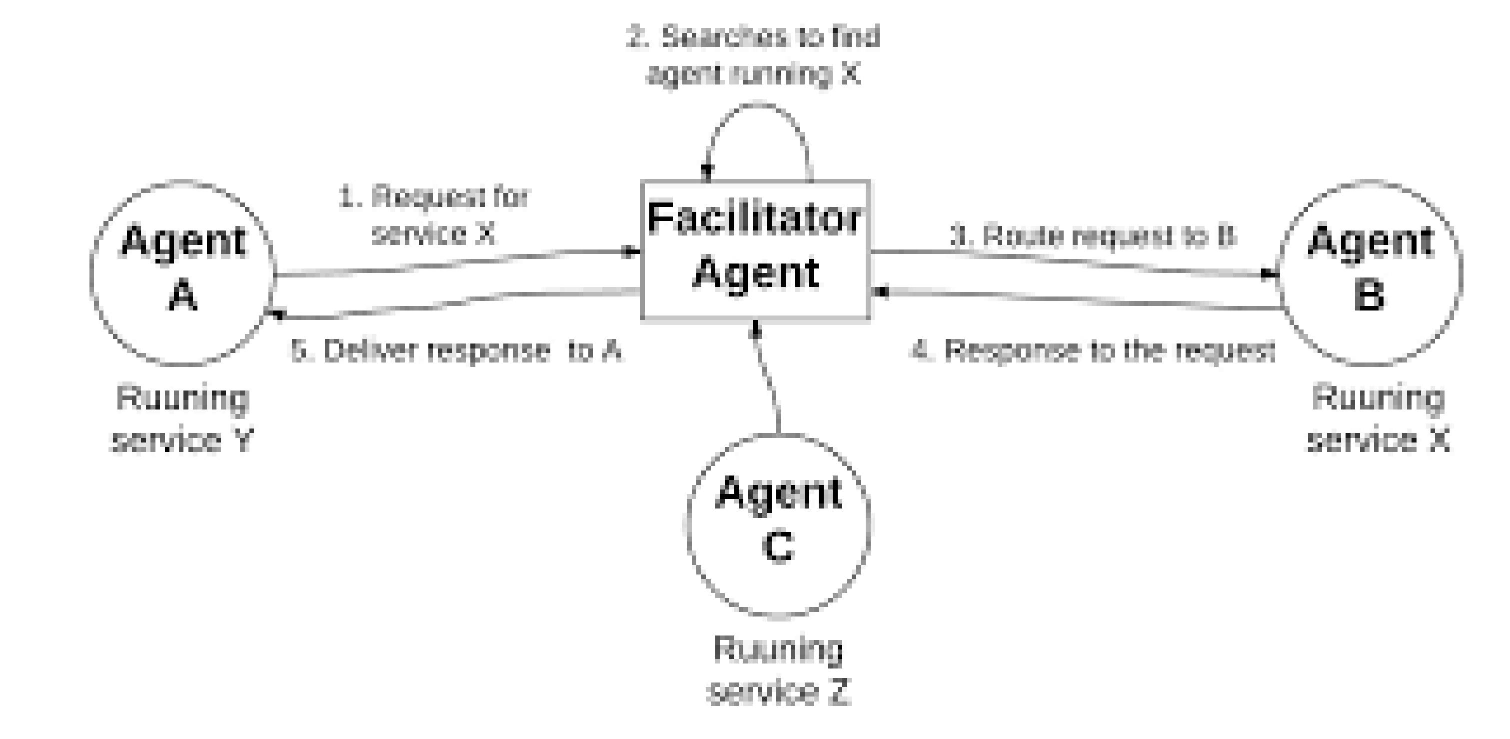
- In other domains (e.g. transportation ecological science or life science), ABM mostly refer to an individual-based model or a self-sufficient computing method.
- Although ABM is a wide ranging paradigm applied in totally different manners in all types of scientific domains, eventually all its subtle differences meet together under the domain of agent-based computing.

AGENT-BASED MODEL (ABM)

- Even though there is a significant overlay, MAS not necessarily means the same as ABM. The objective of an ABM is to search for descriptive insights into the agents’ (not necessarily intelligent) collective behaviour following simple rules (typical of natural systems) rather than solving particular engineering problems.
- ABM is more often used in the sciences, whereas MAS is frequently applied in engineering- and technology-related issues.

AGENT-ORIENTED METHODOLOGY

- Multi-agent systems can be seen as a software development paradigm that provides adequate metaphors to model and implement complex and dynamic software systems. Several agent-based approaches have been proposed to support the development of such systems, including many agent-oriented methodologies.



MAS CHARACTERISTICS

CHAPTER 2

INTRODUCTION

- An agent could refer to different components as applications have different objectives and lay down in different paradigms.
- One can see an agent as being part of a program (e.g. model, system, or subsystem) or any type of independent entity (e.g. organization or individuals). Each agent is programmed to react to other agents and to its computational environment, with respect to behaviour rules from primitive reaction decisions to complex adaptive AI.

CHARACTERISTICS

- Wooldridge and Jennings who defined an agent as a piece of hardware or a software-based computer system that entail the following properties:
  - Reactivity, in the sense agents have the perception of their environment and respond quickly to changes that may occur.
  - Pro-activity, not being limited to acting in response to the environment, agents are able to take the initiative and show behaviour driven by objectives.
  - Social skills. The agents are able to interact/communicate (cognitive model) with other agents (and possibly humans) through a given Agent Communication Language (ACL) and establishing connections between their autonomous objectives and the spatial context.

MAS ARCHITECTURE

CHAPTER 2

MAS ARCHITECTURE

- Agent: An agent is the fundamental building block of a multiagent system. It is an autonomous entity capable of perceiving its environment, making decisions, and taking actions to achieve its goals. Agents can have varying levels of intelligence and complexity, and they may employ different algorithms and reasoning techniques.
- Environment: The environment is the context in which agents operate. It includes all the external factors, entities, and resources that can affect the agents and their actions. Agents interact with the environment to gather information, make decisions, and execute actions. The environment can be physical (e.g., robots navigating a warehouse) or virtual (e.g., software agents in a simulation).

MAS ARCHITECTURE

- Communication: Agents in a multiagent system often need to communicate with each other to share information, coordinate their actions, and negotiate. Communication can be achieved through various means, such as message passing, shared data structures, or direct interaction. Effective communication is crucial for collaboration and coordination among agents.
- Perception: Agents must be able to perceive their environment to gather information necessary for decision-making. Perception can involve sensors, cameras, or other data sources depending on the nature of the agent and its environment. Perception allows agents to update their internal state with information about the external world.

MAS ARCHITECTURE

- Decision-Making: Agents make decisions based on their internal knowledge, goals, and perceptions of the environment. Decision-making can involve various techniques, including rule-based systems, machine learning, optimization algorithms, and heuristic methods. Agents may also employ game theory to make decisions when interacting with other agents.
- Coordination: Coordination is the process by which agents work together to achieve shared goals or resolve conflicts. Coordination mechanisms can range from simple protocols to complex negotiation strategies. Effective coordination is essential to avoid conflicts and ensure that the system's objectives are met efficiently.

MAS ARCHITECTURE

- Agent Architectures: Agents can be designed using various architectures, depending on their complexity and requirements. Some common agent architectures include reactive agents (simple stimulus-response), deliberative agents (plan-based reasoning), and hybrid agents (combining reactive and deliberative elements). The choice of architecture depends on the specific problem domain and the agent's capabilities.
- Goal Management: Agents typically have goals or objectives they aim to achieve. Goal management involves defining, prioritizing, and updating these goals as the situation evolves. Agents may also have mechanisms for handling goal conflicts or revising goals when necessary.

MAS ARCHITECTURE

- Agent Interaction: Interactions among agents can take various forms, such as cooperation, competition, negotiation, and coordination. Agent interaction protocols and mechanisms are designed to facilitate effective communication and collaboration among agents.
- Agent Societies: In multiagent systems, agents are often organized into societies or teams, where they work collectively to achieve higher-level objectives. The structure and roles within these societies can vary widely based on the application domain.

Summary

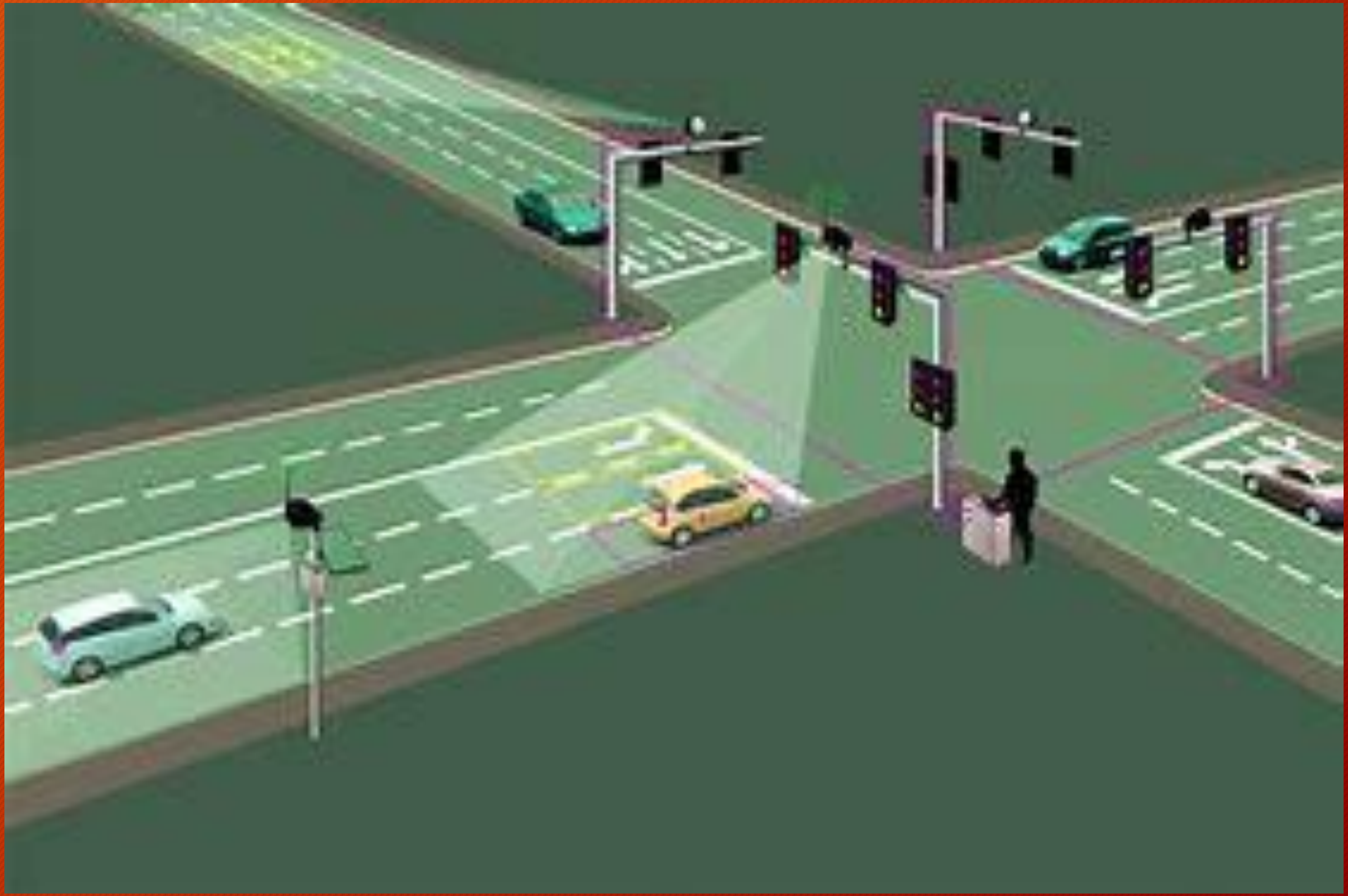
- Overall, the architecture of a multiagent system defines how agents are structured, how they communicate and cooperate, and how they collectively solve problems or achieve goals. Designing an effective multiagent system architecture is a complex task that depends on the specific application and the characteristics of the agents and the environment in which they operate.

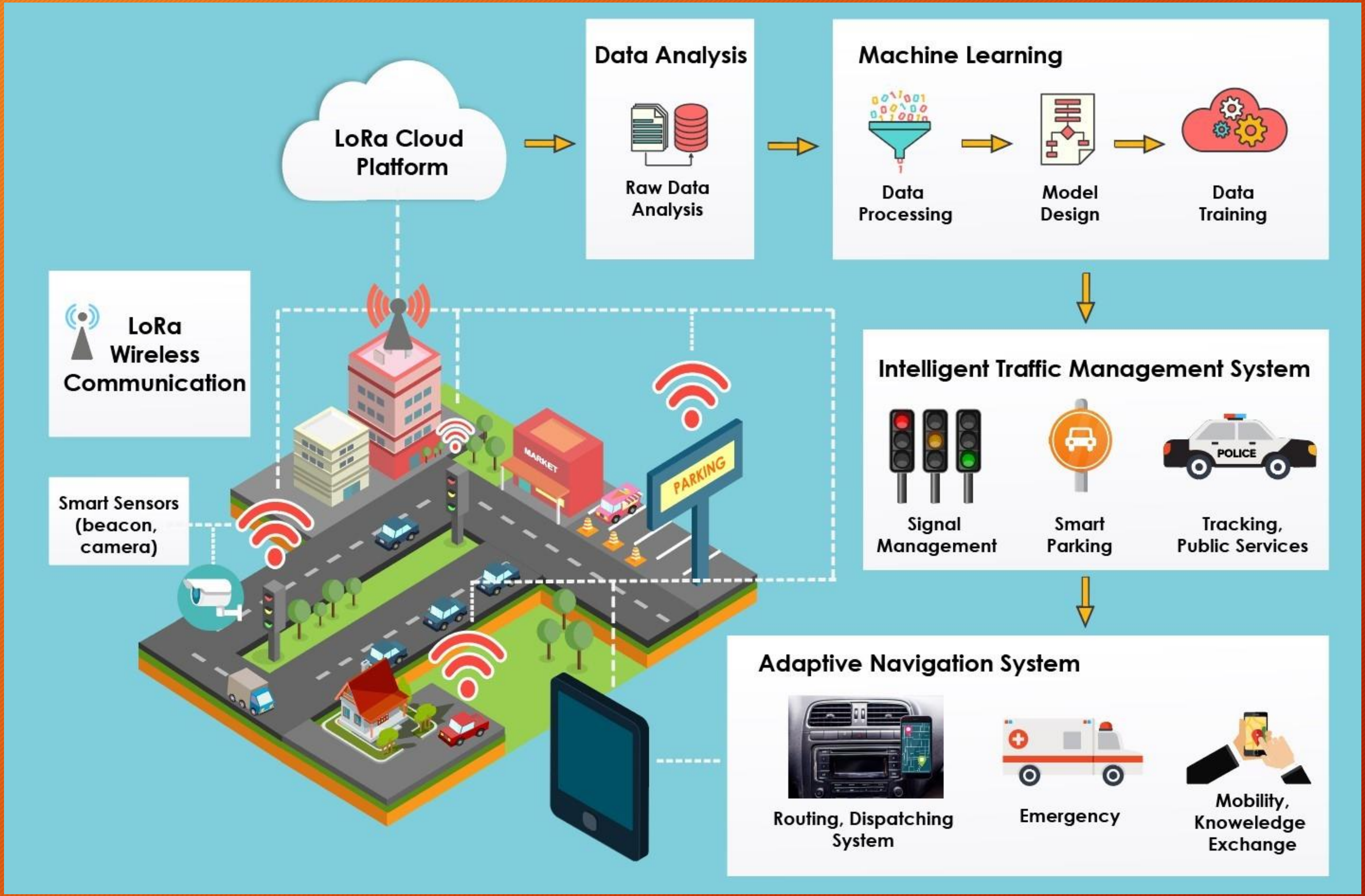
EXAMPLE OF MULTIAGENT SYSTEM

CHAPTER 2

Traffic Management System

- A traffic management system in a smart city is a classic example of a multiagent system. In this scenario, various types of agents interact within a dynamic environment to optimize traffic flow and reduce congestion. Here are the key agents involved:





[Smart Town Traffic Management System Using LoRa and Machine Learning Mechanism – IEEE Future Directions]

Traffic Management System

- **Traffic Lights (Actuator Agents):** These are autonomous agents responsible for controlling traffic signals at intersections. They receive input from various sources, including traffic sensors, cameras, and pre-defined schedules. Based on this information, they make decisions about when to change traffic signals to maximize the flow of vehicles and pedestrians.
- **Vehicle Agents:** Each vehicle on the road is considered an agent in the system. These agents are equipped with GPS and sensors to perceive their surroundings. They communicate with other vehicle agents and the traffic light agents to optimize routes, avoid collisions, and make decisions such as when to accelerate, decelerate, or stop.

Traffic Management System

- **Pedestrian Agents:** Pedestrians are also considered agents in this system. They use mobile apps or wearable devices that provide real-time information about traffic conditions and pedestrian signals. Pedestrian agents can request safe crossing at intersections, and the traffic light agents must consider their requests in the overall traffic control strategy.
- **Central Control Agent:** In some advanced traffic management systems, there is a central control agent that oversees the entire network of traffic lights and collects data from various sources, including traffic cameras and sensors. This agent can adjust traffic signal timings globally to respond to changing traffic patterns or special events.

Traffic Management System

- **Emergency Service Agents:** Ambulances, fire trucks, and police vehicles can be equipped with special agents that request priority access to intersections when responding to emergencies. These agents communicate with traffic light agents to clear the path and ensure quick and safe passage.
- **Traffic Monitoring Agents:** These agents monitor traffic conditions, collect data on congestion, and report incidents such as accidents or road closures. They provide valuable information to the central control agent and help in optimizing traffic management strategies.

Reading a paper

- https://www.sciencedirect.com/science/article/pii/S2352146521003264

ELSEVIER

Transportation Research Procedia

Volume 54, 2021, Pages 918-926

Developing a Traffic Management System Architecture Model

Vadim Korablev [a], Dayana Gugutishvili [a], Aleksandr Lepekhin [a], Berry Gerrits [b]

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.trpro.2021.02.147

Get rights and content

In summary

- In this multiagent system, the agents interact with each other and their environment to collectively achieve the goal of efficient traffic flow, reduced congestion, and improved safety. They make decisions based on real-time data, such as traffic density, vehicle speeds, and pedestrian requests, and adjust their actions accordingly. The coordination and communication among these agents help to create a more responsive and adaptive traffic management system, ultimately improving the overall transportation experience in the city.