Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Bridging the Gap: AI-Driven Agent-Based Systems in Modern Software Engineering

Khulood Abu Maria [1*], Ahmad AA AlKhatib [2]
[1] *Computer Science Department, Al Zaytoonah University of Jordan, Amman, Jordan. Email: khulood@zuj.edu.jo*
[2] *Cybersecurity Department, Al Zaytoonah University of Jordan, Amman, Jordan. Email: Ahmad.Alkhatib@zuj.edu.jo*
**Corresponding Author: Khulood Abu Maria*

ARTICLE INFO

Received: 22 Dec 2024
Revised: 17 Feb 2025
Accepted: 25 Feb 2025

ABSTRACT

Agent-based systems (ABS) are the newest and most effective approach in software engineering to solve complex, chaotic, and interconnected problems. Since ABS models the systems as the agents that communicate with each other, it is an effective approach to creating flexible software. This paper aims to discuss the possibility of applying agent-based systems to software engineering with a focus on the opportunities and challenges and further research directions. This paper presents a conceptual model that demonstrates the connections between the agents, their surroundings, and the software engineering process. The study also creates an AI-Driven Agent-Based System Development Framework (AI-ABSD Framework) that uses machine learning (ML) and artificial intelligence (AI) in the Agent-Based System Development Life Cycle (ABSDLC). This framework was created because of the growing need for very smart, self-guiding, and emotionally intelligent computers. Based on a case study, empirical validation, and comprehensive evaluation, the paper ends with a perspective of the future of ABS in software engineering, highlighting how it can change the approach to developing software systems.

Keywords: Agent-Based Systems (ABS), AI-Driven Framework, Agent-Based System Development Life Cycle (ABSDLC), Machine Learning (ML) in Software Engineering, Artificial Intelligence (AI) Applications, Reinforcement Learning (RL), Emotional Modeling in Agents, Smart Traffic Management Systems, Multi-Agent Systems (MAS), Intelligent Transportation Systems (ITS), Human-Agent Collaboration.

INTRODUCTION

Technology has led to the development of sophisticated software systems that must work in changing, interconnected, and unpredictable conditions. Strict and centrally controlled methods of software engineering, for example, the waterfall, Agile or object-oriented analysis and design model , are not flexible enough to deal with the issues mentioned above. Agent-based systems (ABS) are a viable solution since they present systems as agents that communicate and cooperate to achieve certain objectives. The following are areas or fields that these agents are capable of: smart cities, healthcare, supply chain management, and autonomous systems, among others [1].

Despite the potential of the current ABS methodologies, they are found to have limited implementation of advanced AI and ML techniques, which hampers their capability to address challenging realistic conditions. To fill this gap, this paper introduces the AI-Driven Agent-Based System Development Framework (AI-ABSD Framework), where AI/ML techniques are incorporated into the ABSDLC. This framework makes the agents able to learn from the data, handle changes in the environment, and make better decisions, as well as having emotional models that can improve the human-agent interaction.

This paper has several key contributions, including a conceptual model that depicts the connections between the agents, the context of those agents, and the software engineering process. It also identifies a new framework called the AI-ABSD Framework, where AI/ML techniques and emotional models are integrated into the ABSDLC. This paper also contains a real-world case that shows how the framework can be used. It is also found that this paper is also substantiating the framework with the quantitative assessment of the effectiveness, efficiency, and flexibility of

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

the framework as well as the analysis of the opportunities and risks of the framework and the strengths over the current methodologies.

RELATED WORK

Existing research on agent-based systems has focused on methodologies such as GAIA [2], Tropos [3], and Prometheus [4], which provide structured approaches for designing and implementing ABS. However, these methodologies do not fully leverage the potential of AI/ML techniques to enhance agent behavior. Recent advancements in AI, such as reinforcement learning [5], deep learning [6], and natural language processing [7], have shown promise in improving the adaptability and intelligence of agents. Our work builds on these advancements by integrating AI/ML techniques into the ABSDLC, enabling the development of more intelligent and adaptive agent-based systems.

AI-DRIVEN AGENT-BASED SYSTEM DEVELOPMENT FRAMEWORK (AI-ABSD FRAMEWORK)

The AI-ABSD Framework incorporates AI/ML algorithms and emotional theories into the ABSDLC to offer a systematic way of creating, implementing, and maintaining agent-based intelligent systems. The framework is divided into six phases, all of which involve the use of AI/ML techniques and emotional models to improve the behavior of an agent, as clear in Figure 1.

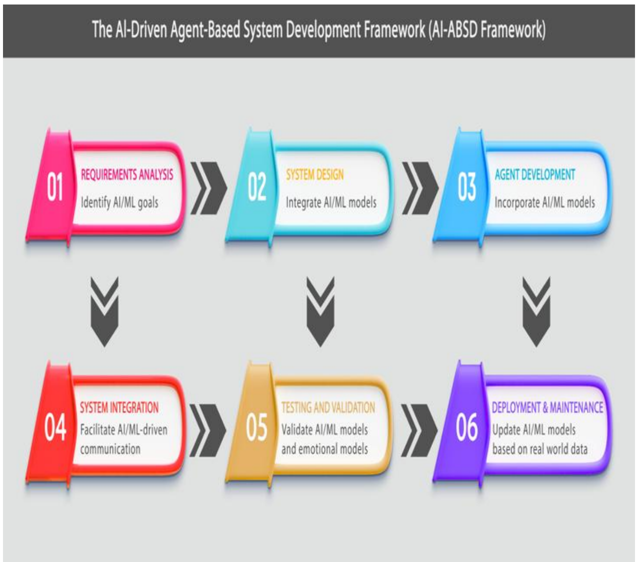

Figure 1: AI-Driven Agent-Based System Development Framework (AI-ABSD Framework)

The AI-Driven Agent-Based System Development Lifecycle Concept (AI-ABSD Framework) presents the six phases of the ABSDLC and the integration of AI/ML algorithms and emotional models in each of the phases.

3.1 Requirements Analysis:

In this phase, the goals, roles and activities of the agents are depicted. In order to achieve this, AI/ML techniques are employed to capture the data sources and learning objectives of the agents. For instance, in a healthcare context, the agents that portray patients may require learning from the patient's data to estimate the health status and feeling happy or anxious according to the state of health [8].

3.2 System Design:

In the design phase, the architecture of the agent-based system is realized. In this paper, AI/ML models are incorporated into the agent architecture that enables the agents to learn from the data and make smart decisions. For example, reinforcement learning models can be applied to shape the agent behavior in the changing conditions [5].

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

3.3 Agent Development:

In this phase, the first is the implementation of individual agents. In this paper, AI/ML models are integrated into the agent's decision-making framework for improving flexibility. For instance, an agent with a deep learning model may organize its tasks according to the level of urgency, for instance, fear or stress, or an agent may learn how to work with other agents in a team in a way that it communicates its trust or empathy [6].

3.4 System Integration:

During the integration process, all agents communicated with each other, allowing them to unite under a single system. The agents were able to apply the knowledge of AI/ML models they had learned, as well as their emotional states. For instance, in a supply chain system, suppliers and manufacturers' agents can employ artificial intelligence-based negotiation techniques to solve conflicts [9].

3.5 Testing And Validation:

During this phase, we test the system to confirm its functionality. To validate the AI/ML models, scenarios are created in which entities display and handle emotions. For instance, a smart traffic management system uses reinforcement learning to assess the agents representing drivers in stressful traffic situations [10][11].

3.6 Deployment And Maintenance:

After deploying the system, the operation is followed by monitoring and maintenance. In this paper, AI/ML models are reinforced with real-world data to enhance the agent's behavior. For instance, in a customer service scenario, the agents can gain knowledge from user interactions to enhance their ability to handle and identify customers' emotional states through natural language processing (NLP) [12].

The Key Features of the AI-ABSD Framework:

- **Structured:** This implies that there is involvement of different groups, such as the stakeholders, developers, and data scientists, in the process to ensure that the goals and objectives of the AI/ML are in consonance with the system.
- **End-to-End Integration:** It encompasses all the phases of the process, starting from setting goals, implementing the system, and installing and supporting it.
- **Focus on Validation:** These are some of the measures that have been put in place to ensure that the AI/ML models and the system are as effective as they are supposed to be.

Scalability and Adaptability: The workflow is also capable of working with complex data and changing data so that the system remains useful and accurate.

INTEGRATION OF URBAN TRAFFIC CONTROL (UTC) WITH AI-ABSD FRAMEWORK

With the fast-expanding number of vehicles in smart cities, managing road intersections and alleviating traffic congestion have become significant challenges. Drivers often express the opinion that dynamic traffic light scheduling, tailored to real-time traffic flows, can greatly improve traffic movement. Integrates a smart traffic control management system termed Urban Traffic Control (UTC) with the AI-Driven Agent-Based System Development Framework (AI-ABSD Framework) to upgrade road traffic network management.

The UTC system incorporates methodologies such as vehicle counting, dynamic lane status evaluation, and adaptive traffic light control to organize traffic across the entire network rather than focusing solely on intersections. The system aims to minimize traffic congestion and reduce the trip and waiting times of vehicles at crossings. Key indicators and models introduced in this study include lane weight, traffic jam indicator, and vehicle priority, which collectively optimize traffic flow with minimal infrastructure changes.

This enhanced system can be implemented on standard traffic lights, allowing for seamless integration with existing infrastructure. It ensures fair movement opportunities for each lane and accounts for no-interference lane movement to further optimize flow. By simulating the system using the NetLogo platform, which enables multi-agent urban

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

traffic modeling, the study tested its effectiveness under real-world conditions. A network of 25 linked intersections with 150 vehicles displaying random behavior was simulated over a 9-hour duration. The proposed UTC system was compared to both fixed-time traffic lights and no-interference movement flow models.

To prove the efficiency of the AI-ABSD framework, we implemented it in an Urban Traffic Control (UTC) with the help of NetLogo, a popular multi-agent simulation tool. We selected NetLogo due to its ease of use, versatility, and ability to create systems with agents capable of mutual interaction [13][14]. The following is a description of the implementation details, the simulation setup, and the results obtained in support of the application of the framework.

4.1 Implementation Details

NetLogo, a multi-agent programmable modeling environment, was used to simulate the urban traffic network. This free, open-source platform employs Java and Scala languages and operates over the Java Virtual Machine (JVM). The reason we chose NetLogo for building the Urban Traffic Control (UTC) using the AI-ABSD framework was because it offers a flexible and modular structure that lets us easily combine different framework parts.

The system was designed to operate in a grid world, where agents (vehicles) navigate through a virtual environment, aiming to reach their respective goals (traffic intersections) while avoiding collisions and optimizing their journey time. We chose the single grid world domain to simplify the simulation and highlight the practical application of the framework.

The simulation model incorporated four agent types: patches, links, turtles, and observers, each playing distinct roles in traffic management. The following components were used in developing the smart Urban Traffic Control (UTC) in NetLogo as shown fin Fig. 1:

- *Traffic Light Agents:* It is responsible for the regulation of traffic signals with the help of real-time traffic information.
- *Vehicle agents:* These are fictions that depict automobiles and are able to report to the traffic signals their location with a view of finding the best routes.
- *Intersection Controller Agents*: Central units ensuring smooth coordination and optimization of traffic light operations.

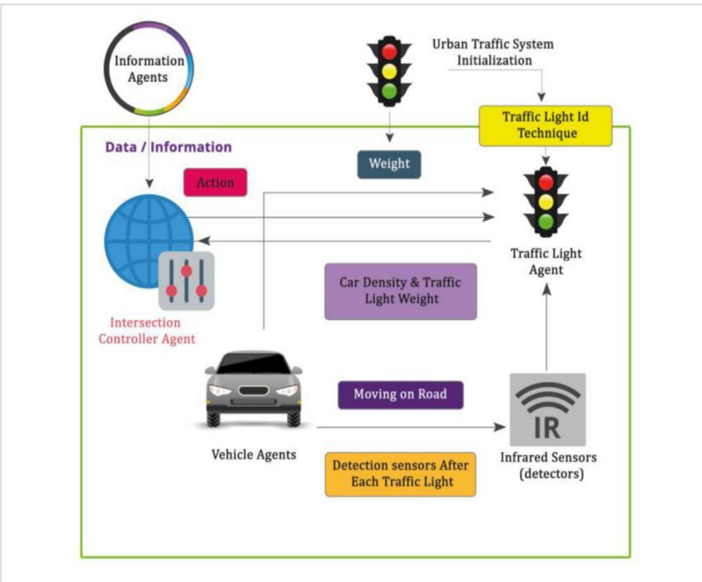

Fig. 1 Multi-Agent Urban Traffic System Components

Table 1 shows each agent's attributes and methods that clarify each agent behavor in the environment.

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1: Agent's main attributes and methods

Agent Type	Attributes	Behavior
Vehicle Agent	• Vehicle Id Priority Direction Current Lane Stopping-Time Waiting-Time Quit-Time	• Set-Id Set-Priority Send-Priority- Signal
Traffic Light Agent	• Traffic-Light-Id Intersection-Id Traffic-Light-Direction Traffic-Light-Movement Total-Vehicles-No Total-Weight	• Set Traffic Id Calculate Vehicle Density Calculate Traffic Light Weight
Intersection Controller Agent	• Intersection-Id Traffic-Light-Id Travel-DirectionTraffic-Jam-Indicator (TJ)Traffic-Jan-index (for movement flow) Selected-Traffic-Light Green-Cycles-NumberYellow-Cycles-Number (10 s as a safety time) Red-Cycles-Number Current-Status Time of Next Status	• Calculate-Traffic-Jam-Indicator Assign-Traffic- Light-Order Calculate- Green-Cycles Calculate-Red-Yellow-Cycles Determine- Next-Change

Proposed AI/ML Models:

- Reinforcement Learning (RL): This one is used by traffic light agents with the aim of improving the signal timing. Some model that can be used in our frame work are:
  - *Q-learning*: A model-free RL algorithm designed to teach agents how to act in a given environment to achieve the highest possible cumulative reward. In ABS, this can be employed to fine-tune the agent's behavior e.g. traffic light control or path planning of an autonomous vehicle.
  - *Deep Q-Networks (DQN):* Combines the idea of Q-learning with deep neural networks to address the issues related to state spaces of high dimensionality. It can be applied to environments with large state spaces, including smart cities or industrial robotics.
  - *Proximal Policy Optimization (PPO):* A well-known policy gradient method for continuous action spaces. By using the PPO algorithm, agents can be trained to exhibit sophisticated behaviors in different environments.
  - *Multi-Agent Reinforcement Learning (MARL)*: Employed when many agents are operating within the same environment and each agent learns from its environment.
- Deep Learning (DL): This one is applied by vehicle agents for the prediction of traffic conditions and for choosing the best paths. Some models that can be implemented in our framework are:
  - *Convolutional Neural Networks (CNNs):* These are mainly used in image classification and processing. They can be used for the evaluation of visual information e.g. from cameras, traffic conditions, pedestrians, or obstacles in autonomous cars.
  - *Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs):* A step up from traditional statistical models that address sequential data or time series.
  - *Generative Adversarial Networks (GANs):* GANs can be used to create realistic data for training ABS in datasets where there is currently little data available Included, for instance, is the creation of artificial traffic patterns or data for learning for agents.

Emotional Models:

In order to make the agents more realistic, they were provided with emotional models that would make the agents behave more like humans. For instance, the vehicle agents could be set to show 'stress' during traffic congestion, while the pedestrian agents could be set to show 'impatience' when they are delayed. Some emotional model that can be applied in our frameword are:

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

- *The OCC Model (Ortony, Clore, and Collins)*: This model is based on cognitive appraisal theory, according to which emotions are the individual's cognitive evaluation of the events in the environment.
- *Plutchik's Wheel of Emotions*: Introduced by Robert Plutchik, this model is one of the most famous psycho-evolutionary theories of emotions. Plutchik argued that eight basic emotions come in pairs, and can be combined to create other emotions.
- *Ekman's Basic Emotions Model*: Ekman proposed a model of six basic emotions that are cross-culturally innate and are manifested through facial expressions.

4.2 Simulation Setup

In the NetLogo simulation, the environment was represented by a grid that contained a number of cities, roads, traffic lights, and crosswalks, and each intersection also had a traffic light agent. The vehicle agents were moved on the roads throughout the different locations between the source and the destination with minimal traffic hold-up. The pedestrian agents were located at the crosswalks, and they were allowed to move through the crosswalks when the traffic lights were on. The simulation's environment is illustrated in Fig. 2.

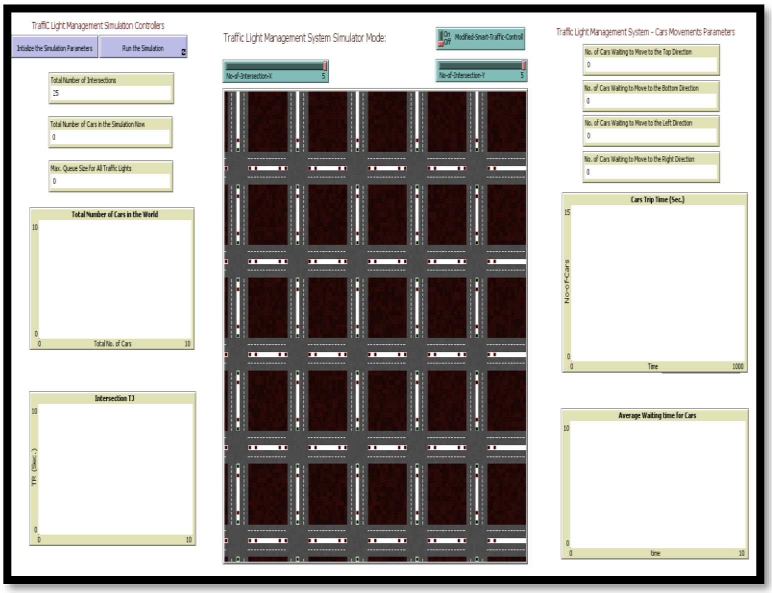

Fig. 2 Simulation Environment Layout [15]

Environment:

- A grid-like structure that depicts a city and its features, such as intersections, roads, and crosswalks.
- Traffic lights were also fitted at every junction, and traffic flow.

Agent Behavior:

- *Traffic Light Agents:* In this paper, traffic signal control has been implemented by using reinforcement learning to vary the signal timings according to the traffic conditions.
- *Vehicle Agents*: These agents employ a deep learning model to assess the traffic situation and choose the best graph.

Simulation Parameters:

The simulation setup included a 25-intersection map with multi-directional lanes and randomly generated vehicle behaviors. Three UTC systems were tested: (1) the proposed dynamic traffic control system, (2) a no-interference movement flow system, and (3) a fixed-time traffic light system.

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

- *Number of Vehicles*: 10,000+ vehicles.
- *Number of Intersections*: 25+ intersections.
- *Simulation Duration*: 24 hours (simulated time).

Data Sources:

- *Real-Time Traffic Data*: Includes information collected by sensors, cameras, and GPS devices that are deployed throughout the city.
- *Historical Traffic Data:* It is applied for the training of the AI/ML models for traffic flow prediction.
- *Weather Data*: To ensure that the model takes into consideration any weather factors that may impact the traffic flow.

4.3 Methodology

The development and implementation of the smart Urban Traffic Control (UTC) using the AI-ABSD framework was done in a very systematic and incremental manner. With this, the approach aim can be attributed to the fact of the objectives of that system, such as traffic congestion, travel time, and pedestrian safety.

The process involved six key phases: these are requirements analysis, system design, agent development, integration, testing and validation, and deployment and maintenance. All the phases were implemented in a very much planned way in order to make the system functional and usable in the real world.

1. Requirements Analysis

- **Goals**: The main goals were to address the problem of traffic jams and enhance the efficiency of the time spent on the road.
- **Agent Roles**: The traffic light agents, the vehicle agents, and the agents were and identified. The traffic light agents managed the signal control; the vehicle agents were involved in the route finding.

2. System Design

- **Architecture:** The base architecture of the system was proposed to incorporate the AI/ML models (for instance, deep learning and natural language processing) and the emotional models.
- **Component Integration:** The implementation makes sure that there is proper coordination between the agents, the environment, and other data sources, including real-time traffic sensors and historical traffic databases.

3. Agent Development

- **Implementation:** The agents will be designed and programmed in NetLogo. The whole system that was AI/ML-written models in Python with the use of libraries like TensorFlow and PyTorch.
- **Behavior Rules:** The traffic light agents' behavior is learned with the help of reinforcement learning (RL) for tuning the signal timings, while the vehicle agents' behavior was predicted with the help of deep learning (DL).

4. System Integration

- **Cohesion:** The agents were integrated into a coherent system of agents by using NetLogo as a multi-agent system.
- **Inter-Agent Communication:** Structures were put in place through which the agents were able to exchange information and data, such as the traffic light agents being able to provide signal timings to the vehicle.

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

5. Testing and Validation

- **Simulation** Testing: The system was also tested in a simulated environment of a congested junction, and the scenarios were created in a way that they are likely to occur in the real-world traffic conditions.
- **Model Validation:** The AI/ML models were tested with real-time traffic flow information to check the effectiveness of the models for predicting traffic flow and for finding the best possible routes.

6. Deployment and Maintenance

- **Pilot Deployment:** This system was implemented in a single city, and the system was assessed under actual conditions of use.
- **Model Updates:** The AI/ML models were occasionally replenished with data collected in the real environment with the purpose of enhancing the models' precision and flexibility to the varying traffic conditions.

4.4 Simulation Discussion

The simulation results of the proposed AI-ABSD framework show that it is applicable in real-life scenarios. Combining AI/ML techniques and emotional models enabled the system to adapt to changes in quality, computing resources, and conditions. These problems made us better at what we did and helped us figure out some ways to make sure that the problems were limited to high-quality data collected during implementation and data that was used to fine-tune AI/ML models for the best performance.

The results demonstrated the effectiveness of the proposed approach, yielding the following outcomes:

- A 25.98% reduction in the total average waiting time for all vehicles over the simulation period.
- A 34.16% reduction in waiting time under the no-interference movement flow scenario.

These results underscore the capability of the proposed UTC system, integrated with the AI-ABSD Framework, to address modern traffic challenges effectively. The system not only reduces waiting times and traffic jams but also minimizes fuel consumption, greenhouse gas emissions, and the environmental impact on urban areas, including flora, fauna, and human health.

DISCUSSION AND ANALYSIS

This section offers a comprehensive breakdown of the AI-ABSD Framework, its strengths, possible issues, and how it is superior to other methods. The AI-ABSD Framework has several key advantages that make it a versatile and powerful framework for managing complex multi-agent systems, as shown in Table 2.

Table 2: Benefits of the AI-ABSD Framework

Benefit	Description
Adaptability	Agents can dynamically adjust their behavior based on real-time data.
Scalability	The framework supports large-scale distributed systems.
Emotional Intelligence	Agents can express and respond to emotions, enhancing human-agent interaction.
AI/ML Integration	Agents leverage AI/ML techniques to make intelligent decisions.

Even though using the AI-ABSD Framework has many potential benefits, there are a few problems that need to be resolved before it can be successfully implemented. The challenges are mentioned in Table 3.

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Table 3: Challenges of the AI-ABSD Framework

Challenge	Description
Complexity	Designing and implementing AI-driven agents requires specialized knowledge.
Data Dependency	The framework relies on high-quality data for training AI/ML models.
Computational Resources	Large-scale systems require significant computational power.
Standardization	Lack of widely accepted standards for AI-driven ABS.

The comparative analysis focuses on the strengths and weaknesses of various agent-based system (ABS) methodologies. As indicated in Table 4, such traditional methodologies as GAIA, Tropos, and Prometheus have a low level of integration with AI/ML models and cannot incorporate emotional models, which limits their effectiveness in dealing with difficult, evolving, and people-oriented applications.

Table 4: Comparative Analysis of ABS Methodologies

Methodology	AI/ML Integration	Emotional Models	Scalability	Adaptability
GAIA	Limited	No	Moderate	Low
Tropos	Limited	No	Moderate	Moderate
Prometheus	Limited	No	High	Moderate
AI-ABSD Framework	High	Yes	High	High

CHALLENGES AND FUTURE DIRECTIONS

There are several challenges that can hinder the adoption of ABS in software engineering:

- *Complexity*: Implementing agent-based systems as well as the design of such systems can be more complicated when compared to the traditional approaches, which can pose the need for certain expertise [2].
- *Lack of Standardization*: This is due to the fact that there are no well-defined standards for AOSE, which makes the systems developed in this manner non-interoperable and unable to be reused [13].
- *Scalability Issues:* This can be problematic in handling large-scale systems, especially in as far as the resource allocation and coordination are concerned [3].

Future Research Directions Include:

- *AI-Driven Agents*: The use of the most recent AI methods, like federated learning and transfer learning, to improve the capabilities of the agents [16].
- *Edge Computing*: Placing agents close to the edge to allow for real-time decision-making in complex systems [17][18].
- *Human-Agent Collaboration:* Improving the communication between humans and agents through implementing natural language processing and human-computer interaction [19].

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

CONCLUSION

Agent-based systems are one of the most effective approaches to the problems of modern software engineering. Through the use of agents, which are autonomous, flexible, and can be scaled, ABS provides a solid platform for the development of large-scale systems. The proposed AI-Driven Agent-Based System Development Framework (AI-ABSD Framework) incorporates AI/ML and emotional models into the ABSDLC to facilitate the creation of intelligent, adaptive, and emotional agent-based systems. The case study, empirical validation, and detailed analysis prove the efficiency of the framework in practical use. However, there are still some issues to be solved, but thanks to the further research and development, the potential of the ABS adoption in software engineering is growing. In the further progress of the agent-based systems concept, one can forecast the revolutionary changes in various fields with the impact on the software development processes.

REFERENCES

[1] M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed. Wiley, 2009.
[2] F. Zambonelli, N. R. Jennings, and M. Wooldridge, "Developing multiagent systems: The Gaia methodology," ACM Transactions on Software Engineering and Methodology, vol. 12, no. 3, pp. 317–370, 2003.
[3] P. Bresciani et al., "Tropos: An agent-oriented software development methodology," Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pp. 203–236, 2004.
[4] L. Padgham and M. Winikoff, Developing Intelligent Agent Systems: A Practical Guide. Wiley, 2004.
[5] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. MIT Press, 2018.
[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[7] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.
[8] R. W. Picard, Affective Computing. MIT Press, 1997.
[9] J. M. Swaminathan et al., "Modeling supply chain dynamics: A multiagent approach," Decision Sciences, vol. 29, no. 3, pp. 607–632, 1998.
[10] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 2, pp. 485–497, 2010.
[11] Hnaif, A.A., Nagham, A.M., Abduljawad, M. and Ahmad, A., 2019, April. An intelligent road traffic management system based on a human community genetic algorithm. In 2019 IEEE Jordan international joint conference on electrical engineering and information technology (JEEIT) (pp. 554-559). IEEE.
[12] V. Dignum, Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way. Springer, 2019.
[13] FIPA, Agent Communication Language Specifications, Foundation for Intelligent Physical Agents, 2002.
[14] Alkhatib, A.A., Maria, K.A., AlZu'bi, S. and Maria, E.A., 2022. Smart traffic scheduling for crowded cities road networks. *Egyptian Informatics Journal*, *23*(4), pp.163-176.
[15] Alkhatib, A.A., Abu Maria, K., Alzu'bi, S. and Abu Maria, E., 2022. Novel system for road traffic optimisation in large cities. *IET Smart Cities*, *4*(2), pp.143-155.
[16] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2020.
[17] W. Shi et al., "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.
[18] Abdel-Fattah, F., Fayyad, S., Heyari, A.M. and Al-Zoubi, H., 2023, August. A Survey of Internet of Things (IoT) Forensics Frameworks and Challenges. In 2023 International Conference on Information Technology (ICIT) (pp. 373-377). IEEE.
[19] P. Bresciani et al., "Tropos: An agent-oriented software development methodology," Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pp. 203–236, 2004.