IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, April 2026
Impact Factor: 8.2

Design of a Priority-Based Adaptive Traffic Signal Control System Using Object-Oriented Principles

Govinda Shinde[1], Ms. Punashri Patil[2], Sujal Singh[3], Yashwant Wankhade[4]
Assistant Professor, Department of Information Technology[2]
Under Graduate Student, Department of Information Technology[1,3,4]
AISSMS's Institute of Information Technology, Pune, Maharashtra, India

Abstract: *Rapid urban growth and the continuous increase in vehicle numbers have intensified traffic congestion in cities. Fixed-time traffic signal controllers do not adapt to real-time traffic flow, resulting in inefficient roadway usage and delayed emergency response times. This paper presents the design and implementation of an adaptive traffic signal control system built upon Object-Oriented Programming (OOP) principles using Java. The system models traffic entities—vehicles, roads, and signal controllers—incorporating abstraction, encapsulation, inheritance, and polymorphism. A priority-based algorithm dynamically adjusts signal timing based on vehicle concentration and emergency vehicle presence. The OOP-based structure facilitates modular development, scalability, and maintainability. Results confirm that a structured software design significantly improves the adaptability of traffic signal control compared to static, fixed-time methods. The proposed model provides a foundation for future Intelligent Transportation Systems and Smart City applications.*

Keywords: Adaptive Traffic Signal Control, Object-Oriented Programming (OOP), Emergency Vehicle Prioritization, Traffic Simulation, Polymorphism

I. INTRODUCTION

High urbanization and rate of car congestion have worsened traffic jams in the urban areas. Traffic control has been central to the problem of sustainable urban development, which has an effect on fuel consumption, environmental effects, and efficiency in emergency response. Traditional types of traffic lights mostly work based on fixed timing programs hence do not have much flexibility regarding instantaneous traffic flow[3].

The current developments in intelligent transportation system underscore the significance of adaptive signal control systems to respond to real time traffic conditions[4]. Nevertheless, in addition to hardware integration, sensing technologies, the underlying software architecture is important to make sure that it is scalable, maintains long-term system development, and maintainability.The approach to forming the development of flexible traffic management solutions that will be able to accommodate future expansion requires a structured and modular design.

Object-Oriented Programming (OOP) offers a solid structure on which the components of complex real-world systems can be modeled using the concepts of abstraction, encapsulation, inheritance, and polymorphism[1][2]. OOP allows the system design of vehicles, roads, and signal controllers to be naturally and easily scalable, since objects can interact with one another. The proposed paper outlines an object-oriented design of adaptive traffic signal control in Java, showing the effectiveness of the structured software engineering principles in improving the design of the intelligent urban traffic facilities.

II. PROBLEM STATEMENT

Traditional traffic signals rely on fixed-time mechanisms that do not adapt to real-time traffic density, leading to poor signal allocation and increased congestion. Such systems also lack a systematic procedure for prioritizing emergency vehicles, potentially hindering urgent response in urban settings. Furthermore, many existing traffic control models are

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, April 2026 Impact Factor: 8.2

deployed without object-oriented structuring, limiting modularity, extensibility, and clear representation of real-world traffic entities.

An evolutionary traffic management framework is therefore required—one that dynamically responds to traffic conditions and applies core OOP concepts to create a scalable, integrated structure of vehicles, roads, and signal controllers.

III. LITERATURE REVIEW

A. Bjarne Stroustrup: What is Object-Oriented Programming

Stroustrup's foundational work explains the core features of OOP: classes, inheritance, data abstraction, and polymorphism. His work demonstrates how these features enable scalable system modeling and reduce complexity in large software applications. This source provides the theoretical basis for applying encapsulation, abstraction, inheritance, and polymorphism in the adaptive traffic signal system [1].

B. Peter Wegner: Concepts and Paradigms of OOP

Wegner introduces a formal differentiation between object-based and fully object-oriented systems, emphasizing the significance of inheritance and dynamic behavior in scalable systems. His classification framework is used in this work to justify the use of inheritance and polymorphism for modeling the traffic management system [2].

C. Mirchandani and Head: Real-Time Traffic Signal Control Systems

Mirchandani and Head investigated adaptive traffic signal control systems that respond to actual traffic conditions, emphasizing dynamic decision-making in intersection management. Although their focus is algorithmic, their work justifies the necessity of adaptive signal allocation mechanisms incorporated in this system [3].

D. Intelligent Transportation Systems (ITS) Research Overview

Studies in ITS focus on responsive and adaptive traffic control to enhance congestion management, primarily through optimization techniques and sensor-based integration. This work builds on these adaptive concepts while placing a deeper focus on structured OOP modeling for system scalability and maintainability [4].

IV. METHODOLOGY

A. System Modeling

The proposed adaptive traffic signal control system is modeled by representing real-world traffic entities as structured objects. The system simulates an intersection with multiple roads, vehicles of different categories, and signal controllers. Each road maintains its own vehicle count, and each vehicle type is assigned a priority level that influences signal allocation decisions.

The modeling approach relies on real-world abstraction, where physical traffic elements are mapped to logical classes [1]. Vehicle types include Car, Bus, and Ambulance, enabling differentiated behavior within a shared structural framework. This object-based modeling captures real traffic dynamics while maintaining modular organization.

B. OOP-Based Architecture

The system architecture is designed using core OOP principles. An abstract class serves as the base, defining shared attributes and a priority evaluation method. Derived classes (Car, Bus, Ambulance) extend this base class and override the priority method, demonstrating inheritance and runtime polymorphism. This hierarchical structure allows new vehicle categories to be introduced without affecting existing logic.

Encapsulation is implemented by restricting direct access to critical data such as vehicle count and signal status. The Road and Signal classes manage their states through controlled public methods, ensuring data integrity. The TrafficController class coordinates between roads and signals. This hierarchical design increases scalability, modularity, and maintainability [2].

C. Adaptive Algorithm

The adaptive signal allocation mechanism operates through a priority-based evaluation process. For each road, a priority score P_i is computed using the following formula:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, April 2026 Impact Factor: 8.2

$P_i = (V_i \times W) + E_i$

where:

V_i = number of vehicles on road

W = predefined traffic weight constant

E_i = emergency vehicle priority factor.

The algorithm performs the following steps:

Retrieve vehicle count for each road

Detect the presence of emergency vehicles

Compute priority scores

Compare computed values

Assign the green signal to the road with the highest score.

The selection process operates with time complexity O(n), where n represents the number of roads at the intersection[2].

D. Implementation Details

The system is developed in Java as a console-based simulation model. Implementation focuses on modular class design, with each class having a clear responsibility. Vehicle and road objects are initialized by constructors, and static variables monitor vehicle statistics globally.

Polymorphism is illustrated through dynamic method overriding of the priority evaluation function. Method calls manage object interaction between TrafficController, Road, and Vehicle classes. The modular design enables adding new vehicle types or traffic rules with minimal restructuring. The implementation prioritizes clarity, extensibility, and conformance to OOP principles, making it suitable for simulation analysis and research-scale expansion.

V. SYSTEM DESIGN

A. UML Class Diagram

The structure of the adaptive traffic signal control system is illustrated in Fig. 1.

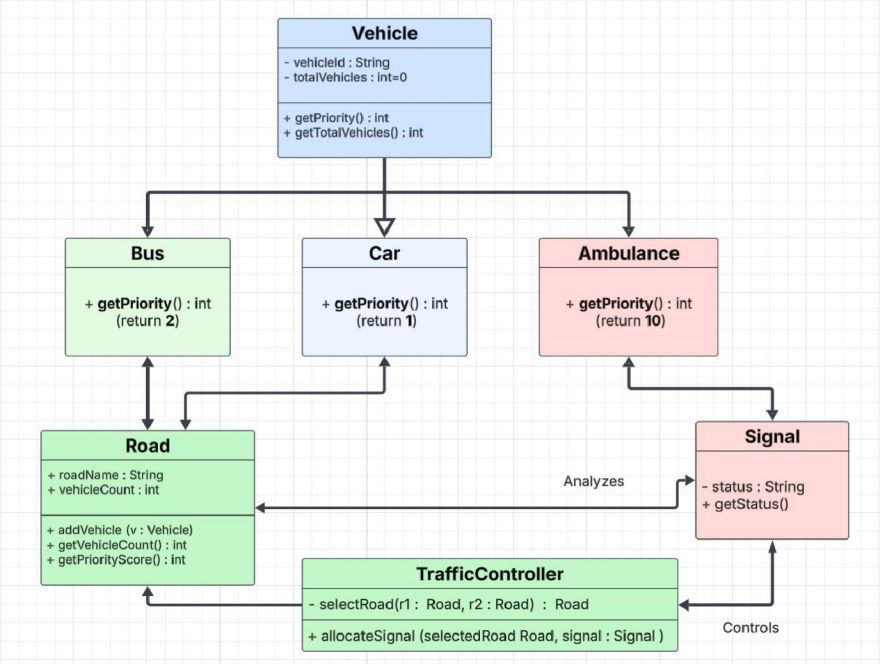

Fig. 1. UML class diagram of the adaptive traffic signal control system.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, April 2026 Impact Factor: 8.2

The diagram demonstrates the inheritance relationship between Vehicle and its subclasses (Car, Bus, Ambulance), the association between Road and Vehicle, and the coordination between TrafficController, Road, and Signal.

B. Application of Object-Oriented Principles

Abstraction is achieved through the abstract class Vehicle, which defines a common interface for priority evaluation without exposing implementation details.

Inheritance enables hierarchical specialization. Vehicle has Car, Bus, and Ambulance as subclasses that inherit shared attributes, allowing code reuse and easy addition of new vehicle types.

Polymorphism is implemented via overriding of the getPriority() function. At runtime, the system selects the correct method for the actual vehicle type, ensuring ambulances receive the highest priority.

Encapsulation maintains the privacy of traffic information—such as vehicle count and signal status—allowing external access only through public methods, ensuring data integrity and preventing unintended modifications.

VI. RESULT AND ANALYSIS

A. Code Implementation in Java

The complete Java implementation of the adaptive traffic signal control system is presented below:

```
// Abstract Base Class (Abstraction)
abstract class Vehicle {
protected String vehicleId;
protected static int totalVehicles = 0;
public Vehicle(String vehicleId) {
this.vehicleId = vehicleId;
totalVehicles++;
}
public abstract int getPriority();
public String getVehicleId() { return vehicleId; }
public static int getTotalVehicles() { return totalVehicles; }
}

// Inheritance + Polymorphism
class Car extends Vehicle {
public Car(String vehicleId) { super(vehicleId); }
public int getPriority() { return 1; }
}
class Bus extends Vehicle {
public Bus(String vehicleId) { super(vehicleId); }
public int getPriority() { return 2; } // Method Overriding
}
class Ambulance extends Vehicle {
public Ambulance(String vehicleId) { super(vehicleId); }
public int getPriority() { return 10; } // Highest priority
}

// Encapsulation
class Road {
private String roadName;
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, April 2026 Impact Factor: 8.2

```
private int vehicleCount;
private int emergencyFactor;
public Road(String roadName) {
this.roadName = roadName;
this.vehicleCount = 0; this.emergencyFactor = 0;
}
public void addVehicle(Vehicle v) {
vehicleCount++;
emergencyFactor += v.getPriority();
}
public int getVehicleCount() { return vehicleCount; }
public int getPriorityScore() { return vehicleCount + emergencyFactor; }
public String getRoadName() { return roadName; }
}

// Signal Class
class Signal {
private String status = "RED";
public void changeSignal(String newStatus) { status = newStatus; }
public String getStatus() { return status; }
}

// Controller Logic
class TrafficController {
public Road selectRoad(Road r1, Road r2) {
if (r1.getPriorityScore() > r2.getPriorityScore()) { return r1; }
else { return r2; }
}
public void allocateSignal(Road selectedRoad, Signal signal) {
signal.changeSignal("GREEN");
System.out.println("Green Signal Allocated to: "
+ selectedRoad.getRoadName());
}
}

// Main Class
public class SmartTrafficSystem {
public static void main(String[] args) {
Road road1 = new Road("Road A");
Road road2 = new Road("Road B");
road1.addVehicle(new Car("C1"));
road1.addVehicle(new Bus("B1"));
road1.addVehicle(new Car("C2"));
road2.addVehicle(new Car("C3"));
road2.addVehicle(new Ambulance("AMB1"));
System.out.println("Total Vehicles: "
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, April 2026 Impact Factor: 8.2

```
+ Vehicle.getTotalVehicles());
TrafficController controller = new TrafficController();
Signal signal = new Signal();
Road selected = controller.selectRoad(road1, road2);
controller.allocateSignal(selected, signal);
System.out.println("Signal Status: " + signal.getStatus());
}
}
```

B. Execution Scenarios

The system was evaluated under three traffic conditions to verify accuracy of decision-making and emergency prioritization behavior:

Scenario 1 — Balanced Traffic: Road A: 3 Cars; Road B: 3 Cars.

Scenario 2 — Uneven Congestion: Road A: 8 Vehicles (Cars + Bus); Road B: 3 Vehicles.

Scenario 3 — Emergency Vehicle Presence: Road A: 5 Cars; Road B: 2 Cars + 1 Ambulance.

C. Output Observation

TABLE I: EXECUTION SCENARIO RESULTS

Scenario	Road A Score	Road B Score	Selected Road	Reason
Scenario 1	3	3	Road A	Equal traffic load
Scenario 2	8	3	Road A	Higher vehicle density
Scenario 3	5	12	Road B	Emergency priority override

D. Observed Limitations

While the system demonstrates effective adaptive behavior, several limitations were identified during evaluation:

• The model is a console-based simulation without real-time sensor input.

• Signal timing duration is not adjusted dynamically.

These limitations indicate potential areas for future expansion.

VII. CONCLUSION

Urban areas are experiencing rapid growth with increasing vehicle numbers, resulting in intensified traffic congestion. Traditional fixed-time traffic signals are unable to adapt to actual traffic conditions, causing poor road utilization and delays for emergency vehicles.

This paper presented an adaptive traffic signal control system developed in Java using OOP principles. The system simulates vehicles, roads, and traffic signals through abstraction, encapsulation, inheritance, and polymorphism. A priority-based algorithm dynamically modulates signal timing based on vehicle density and emergency vehicle presence.

The OOP-based structure ensures the system is modular, scalable, and maintainable—making it well-suited for real-world traffic scenarios. Results demonstrate that structured software design offers significantly greater adaptability than traditional fixed traffic control methods. The proposed model can serve as a foundation for future Smart City and Intelligent Transportation System projects.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, April 2026
Impact Factor: 8.2

ACKNOWLEDGMENT

The authors would like to thank the Department of Information Technology, AISSMS's Institute of Information Technology, Pune, for providing the academic environment and resources that supported this work. Special thanks are extended to the faculty members whose guidance and feedback contributed significantly to the development and refinement of this study.

REFERENCES

[1] B. Stroustrup, "What is Object-Oriented Programming?" in Proc. ACM Conf. Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), 1991.
[2] P. Wegner, "Concepts and paradigms of object-oriented programming," ACM SIGPLAN Notices, vol. 21, no. 10, pp. 168–182, Oct. 1986.
[3] P. Mirchandani and L. Head, "A real-time traffic signal control system: Architecture, algorithms, and analysis," Transportation Research Part C: Emerging Technologies, vol. 9, no. 6, pp. 415–432, 2001.
[4] IEEE Intelligent Transportation Systems Society, "Overview of intelligent transportation systems," IEEE ITS Society. [Online]. Available: https://www.ieee-itss.org/