Intelligent Agent Oriented Architecture for Traffic Management

A Thesis Presented to

Department of Computer Science & Software Engineering
Faculty of Basic & Applied Sciences

In Partial Fulfillment

of the requirement for the degree

Of

Master of Sciences (Software Engineering)

By

Sadia Afsar
(153-FBAS/MSSE/F07)

Supervised by
Dr. Fahim Arif
MCS, NUST

International Islamic University Islamabad
(2011)

1. Computer science
2. Automatic data processing

Department of Computer Science & Software Engineering

International Islamic University Islamabad

Final Approval

Dated: 20 Feb 2012

This is to certify that the Department of Computer Science and Software Engineering, International Islamic University Islamabad, accepts this dissertation submitted, in its present form as satisfying the dissertation requirements for the degree of MS (SE).

Committee

Supervisor:

Dr.Fahim Arif
Assistant Professor,
Department of Computer Software Engineering
NUST
Military College of signals

(Dr.Fahim Arif)

External Examiner:

Dr.Awais Majeed
Assistant Professor,
Department of Computer Software Engineering
Military College of Signals
National University of Science and Technology(NUST)

(Dr. Awais Majeed)

Internal Examiner:

Miss Salma Imtiaz
Assistant Professor,
Department of Computer Science & Software Engineering,
International Islamic University, Islamabad

(Miss Salma Imtiaz)

Acknowledgement:

I would like to acknowledge many people who have supported, guided and advised me during the MSSE thesis. Especially I am grateful to my Professor Dr. Fahim Arif, Department of Computer Software Engineering, Military College of Signals, NUST. His valuable suggestions and motivation helps me to complete my thesis. He always guided whenever I stuck and encourage me during the whole period. I am also thankful to Professor Dr. Navid Ikram for his useful suggestions and great ideas that helped me to move in right direction.

Declaration:

I hereby declare and affirm that this thesis neither as a whole no as part thereof has been copied out from any source. It is further declared that I have completed this thesis entirely on the basis of my personal effort, made under the sincere guidance of my supervisors. If any part of this report is proven to be copied our or found to be a reproduction of some other, we shall stand by the consequences. No portion of the work presented in this report has been submitted in support of an application for other degree or qualification of this or any other University or Institute of learning.

Sadia Afsar

153-FBAS/MSSE/F07

Dedication:

This work is dedicated to my parents and family who always encourage, support and guide me to achieve this milestone.

Sadia Afsar

Abstract:

Emerging trends in software development have changed due to the huge amount of data, growth of internet, mobile, dynamic and smart applications. Most of such applications consist of small, intelligent, flexible and distributed components known as agents. Number of agent methodologies has been presented but few of these are evaluated and verified. Due to the invention of agent technology, the way to analyze, design and build the systems has been changed. Vehicle traffic management especially in large cities is rapidly becoming one of the major challenges due to heavy growth in population and vehicles. The problem of traffic jam occurs when traffic demand is greater than specified time or mismanagement of specified interval for the signals or less available roads and their capacity. To avoid traffic congestion, number of methods has been used from the last many years such as increase number of roads and their size, limiting the number of vehicles, enhancement in current systems etc.

This research proposes an Agent-Based Autonomous Controller (ABAC) architecture for managing city traffic. It uses time series of historical traffic intensity to estimate the appropriate time allocation for each signal at a given intersection. The architecture takes care of the exceptional appearance of rescue vehicles (e.g., ambulance) in order to ensure a smooth flow of the traffic. The ABAC architecture counts on several AI techniques germane to assessing the intensity of the traffic using image recognition algorithms. It also counts on environment sensors (sound sensors) in order to detect the advent of emergency vehicles. The ABAC architecture shows a high degree of adaptability leading to the least need for human intervention. The proposed solution for traffic management is validated through simulation. The simulation results reveal the performance and effectiveness of the ABAC architecture over previous architectures.

Achievements and Contribution:

Journal Publications

1. Smart Traffic Management System, Journal: Journal of Computing (Impact Factor 0.2), Volume 3, Issue 6, June 2011, ISSN 2151-9617, 2011.
2. An Agent-Based Autonomous Controllers for Traffic Management, Lecture Notes of Computer Science (LNCS), 2011.

Conference Publications

1. "Autonomous Agent Oriented Traffic Control Systems", The 2nd International Conference on Computer and Automation Engineering, Singapore, 2010.
2. "An Agent-Based Autonomous Controllers for Traffic Management", 2nd International Conference on Software Engineering and Computer Systems, Pahang, Malaysia, June 27-29, 2011.
3. "Intelligent Controller for Urban Traffic", International Conference on Computing and Information Technology, Madina, Saudi Arabia, March 12-14, 2012.

Table of Contents

Chapter 1: Introduction & Background 1
1.1 Traffic Management System: 2
1.2 History: 3
1.3 Traffic Lights: 4
1.4 Classification of Traffic Signal Controller: 5
1.5 Use of Computers in Traffic 6
1.6 Procedural to Agent Oriented Programming 6
1.7 Statement of the Problem 8
1.8 Research Question 8
1.9 Motivation 9
1.10 Application of the Research 10
1.11 Organization of the Thesis: 11
Chapter 2: Related Work 12
2.1 State of the Art in Traffic Management 12
Chapter 3: Intelligent Agent Oriented Architecture For Traffic Management 16
3.1 The ABAC Traffic Management Architecture: 17
3.2 ABAC Algorithms 22
3.2.1 FullRoundThreshold Algorithm 22
3.2.2 CountVehicles Algorithm 23
3.2.3 SoundAlert Algorithm 25
Chapter 4: System Design 27
4.1 ABAC Architecture: 28
4.2 Use Case Diagram 28
4.3 Class Diagram 31
4.4 State Chart Diagram 32
4.5 Interaction Diagram 33
4.5.1 Sequence Diagram 33
4.5.2 Collaboration Diagram 34

Chapter 5: Dynamic Time Allocation through ABAC Architecture ... **35**
5.1 Classification of Signals ... 36
5.1.1 Independent Traffic Signals ... 36
5.1.2 Mutual-Dependent Traffic Signals ... 41
Chapter 6: Implementation, Results & Conclusion ... **46**
6.1 Introduction to Traffic Simulation ... 47
6.2 Traffic Simulation Entities ... 47
6.3 Independent Traffic Signal ... 48
6.4 Independent Traffic Signal with Rescue Vehicle ... 53
6.5 Mutual-Dependent Traffic Signals ... 55
6.6 Interconnected Traffic Signals with Rescue Vehicle ... 56
Conclusion ... 59
Future Work ... 59
References ... **60**

List of Figures

Figure 1. A simple Traffic Scenario 2
Figure 2. First Traffic Signal installed at London 3
Figure 3. Traffic Signal Light Sequence 5
Figure 4. Abstract view of the ABAC Architecture 19
Figure 5. Architecture of the ABAC System 24
Figure 6. FullRoundThreshold Algorithm 25
Figure 7. CountVehicle Algorithm 26
Figure 8. SoundAlert Algorithm 27
Figure 9. Use case diagram for Sensor (Camera) 29
Figure 10. Use case diagram for Sensor (Sound Detector) 30
Figure 11.Use case diagram for Actuator 30
Figure 12.Class diagram for ABAC 31
Figure 13. State Chart diagram for Signal 32
Figure 14. State Chart diagram for Signal Lights 32
Figure 15. Sequence diagram for ABAC 33
Figure 16. Collaboration diagram of ABAC 34
Figure 17. Average Waiting Time (ABAC Vs Static Traffic Arch.) 40
Figure 18.Mutual-Dependent Traffic Signal Scenario 41
Figure 19. Graph representation of Mutual-Dependent Traffic Signal 42
Figure 20. Independent Traffic Signal Scenario 1 50
Figure 21. Graph of Independent Traffic Signal Scenario 1 51
Figure 22. Independent Traffic Signal Scenario 2 52
Figure 23. Graph for Independent Traffic Signal Scenario 2 52
Figure 24. Independent Traffic Signal Scenario 3 53
Figure 25. Graph for Independent Traffic Signal Scenario 3 54
Figure 26. Independent Traffic Signal with Rescue vehicle Scenario 55
Figure 27. Rescue Vehicle in Independent Traffic Signal 55
Figure 28. Dependent Traffic Signal Scenario 56
Figure 29. Dependent Traffic Signal with Rescue vehicle Scenario 57

List of Tables

Table 1 Comparison b/w ABAC & Static Traffic Architecture 41
Table 2 Signals with Associated Dependency: 44
Table 3 Assigned Time for Mutual-Dependent Traffic Signals 45
Table 4 Results of ABAC Architecture........ 46
Table 5 Results of Conventional Static Traffic Management Architecture 46
Table 6 Results of History Based Traffic Architecture with 100 sec cycle 47
Table 7 Results of History Based Traffic Architecture with 70 sec cycle 47

Chapter 1: Introduction & Background

1.1 Traffic Management System

Traffic can be defined as the aggregation of vehicles that come out or in to some place for a particular event or time. Roads, vehicles and traffic signals are the most important players in any traffic management system. Roads are highways that intersect each other at some specific point. Vehicles are the moving objects such as cars, buses, etc.

Traffic jam is a particular condition on the roads. Generally increase of vehicles increases the queuing and traffic jam. The problem of traffic jam occurs when traffic demand is greater than specified time or mismanagement of specified interval for the signals or less available road capacity. A scenario of traffic on the signal is shown in Figure 1. Traffic management is a way to find the time and some suitable method to arrange the vehicles and tracking of the traffic network problems.

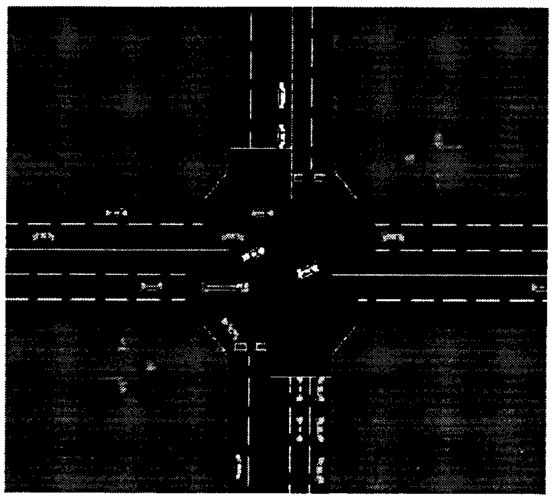

Figure 1. A simple Traffic Scenario

Traffic jam is a common problem around the world and it incessantly imposes challenge on researchers to find effective solutions. At a given intersection point, an effective traffic management should give sufficient time for each vehicle to pass without causing deadlock or backlog. Traffic jam occurs due to interrelated real world problems like wrong lane turns, sudden and unauthorized pedestrian crossing, speed lane violations, police check posts, the operation of emergency and ambulatory services etc.

Normally traffic is controlled by the human being or automatic system. In Human controlled systems, human administrator monitors the incoming traffic and controls the signals while in automatic system a constant time is allocated for each side of the signal. In human operated system, the response time of human administrator is slow due to the large number of vehicles or roads. Automatic traffic system follows the constant time allocation that causes wastage of time and traffic congestion.

1.2 History

Traffic signals were first introduced by J. P. Knight on December, 1868 and it was installed at the British Parliament, London as shown in Figure 2. The signal was introduced with red and green lights only, while red and green lamps were used for night. At that time a lever is used to turn ON the appropriate light. However, the signal failed on January, 1869 by injuring the operator who was responsible to move the lever.

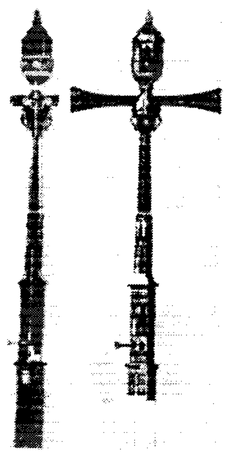

Figure 2. First Traffic Signal installed at London

American policeman, Lester Wire gave the concept of modern traffic signals in 1912 with red and green electric traffic signals. Later on, in 1914 an American traffic signal company put in the traffic signal in Ohio State. The signal was consisting of red, green lights and a buzzer that was fixed to warn the color. Interconnected traffic signal was manually operated in 1917 with six intersection points. The 4-way traffic signal was introduced by the Willium Pots in 1920. The first interconnected traffic signal was introduced in 1917 and installed at Salt Lake City, USA with six junctions and operated through human being. The interconnected signal was automated in1922 at Houstan, Texas. At that time first patent about the traffic was also published with title

"Combination traffic guide and traffic regulating signal". In 1923, the Morgan traffic signal was introduced having T shape pole unit. It has three eccentric positions and operated from a distance manually. The concept of timer was introduced in 1999 at Taiwan. The purpose of the timer is to inform drivers or walkers about the (closing or opening) remaining time.

1.3 Traffic Lights

The traffic lights are used to control the traffic and also known as the semaphores, traffic signals or stoplight. These are fixed at the junction of roads so that the traffic of each side can cross the junction smoothly. Traffic lights use three universal colors i.e. red, green and yellow or Orange. The green color shows that the traffic can proceed, red color shows that the traffic is not allowed to move while yellow or orange is used to alert drivers before the opening or closing of signal.

Traffic signals are the central points where the vehicles can change or remain in the same direction. A vehicle at any given signal can move in three directions, i.e. left, right and straight (Mueller, 1970). The vehicles on a signal are controlled through three lights or a single light having three states i.e. red, green and yellow. Signal light changes according to following four rules that is also represented in Figure 3.

- Green to yellow
- Red to yellow
- Yellow to green
- Yellow to red

There is no direct conversion of lights from green to red and vise versa.

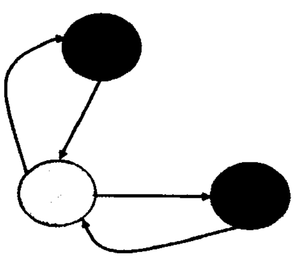

Figure 3. Traffic Signal Light Sequence

1.4 Classification of Traffic Signal Controller

Vehicle traffic is usually probabilistic due to two factors; which are inconsistency in number of drivers, type and size of vehicles while the second is certain special situations like bad weather or some special event. When there is contest among the vehicles to use the same road space at a signal, then there is a need of some set of rules and regulations to avoid traffic congestion, delays and accidents. There are three ways to control the traffic signals, i,e, Fixed, Actuated and adoptive.

1.4.1 Pre-Timed Traffic Signal

One way is to keep a signal in a fixed schedule on the basis of heuristics and historical data. In this type of controller a particular fixed time is allocated to each side and opening sequence should also be fixed. Same sequence or pattern repeated throughout the day or sometime different fixed time allocated at different spans of the day. However the technique has serious limitations such as order and allotted time cannot be changed according to the number of vehicles. Moreover, there will be high average waiting time of vehicles during the night time or when there is low traffic.

1.4.2 Actuated Traffic Signal

To overcome the cons of Pre-time traffic signal, another way is actuated control. It was introduced where the vehicle detectors or push buttons are fixed. They provide information to actuator that takes action accordingly by allocating the particular time according to the traffic flow on each side. In this method the vehicle detectors or walker push button are used to provide traffic information to the controller. There is no mechanism to entertain emergency or any other rescue vehicle. Another problem with actuated traffic signal is the lack of learning capability.

1.4.3 Adaptive Traffic Control Signal

Third or more modern way to handle the traffic is an adaptive control that has the capability to assign the time according to the current traffic flow and continuously learns. It enhances the traffic plans according to the current traffic flow and experiences. The adaptive traffic control consists of three sub components which includes Detection, Prediction and Optimization. The detection is used to detect the traffic flow, the prediction module is responsible to predict the time length while the Optimization module is responsible to check weather there is some enhancement of predicted time or not.

1.5 Use of Computers in Traffic

Use of computers in traffic control (air, ships, and vehicle) system has been increased due to the powerful computation of processor-based systems. It also becomes essential as population and vehicles growth is rapidly rising. All these factors throw in to automate the traffic control system. The automation manages the traffic control itself without the involvement of human beings; which result in reducing the workload with avoiding congestion and increases the traffic flow. Number of systems has been automated through agent oriented and autonomic computing approach. IBM introduced the concept of autonomic computing (Horn., 2001). The aim of this technology is to develop such systems that perform action and manage itself without any external input or support from the humans. Basic characteristics of autonomic computing are self-optimization, self-inspection, self-configuration, self-healing, self- organization and self-protection (Horn, 2001, Kephart, et al., 2003). In 1950 Turing introduces concept of artificial intelligence and agents programs (Markus, et al., 2008). Turing gave the idea that how intelligence can be incorporated in machines. He also proposed that how learning machine can be built and how teaching ability can be integrated. Agents have some special characteristics such as mobility, autonomous, adoption, goal oriented and interaction with other agents or systems to do some useful tasks.

1.6 Procedural to Agent Oriented Programming

In old days, software or programs are developed in computer languages (Pascal, Fortran, Cobol) through data structures and procedures. After that the concept of object oriented programming (OOP) is introduced where a single structure is made through data and function. In OOP, the data is variable instances while the function is called a method. Due to this combination, object can be replicated and reuse easily with self-interest characteristic. There are number of software engineering paradigms such as Procedural, structured, Object oriented, design patterns, agent oriented etc. These paradigms are developed with the passage of time to make software engineering process easier and to handle complexity. Agent oriented techniques have enhanced the capability of analysis, modeling, designing and implementing the distributed systems. They can enhance the software engineering paradigms and applications. Moreover, agent oriented approach is flexible and can divide the complex problems in an efficient way.

The agent technology is actually the extension of component based technology with more interactive and dynamic nature (Stuart, et al., 2008). Agents are self-contained and independent systems that perform different actions to perform predefined tasks. In agent oriented systems, the objective is achieved by executing one part of the system and interacts with other parts. The execution and interaction behavior of the agent is observed to get its objective and enhancement. One of the advantages of this technology is the parallel execution of numerous numbers of agents on a single machine. Analytic model was used before the invention of agent model where the systems are functions and represented through mathematical notations. Analytic models are good as the function behavior is represented as a mathematical equation. However these models are static and do not provide complete preview or visual representation and are mathematically intractable.

The multi-agent systems are presented first time in the first international conference on multi-agent systems (ICMAS-95) at San Francisco in June 1995. The Foundation for Intelligent Physical Agents (FIPA) defines agent as a computational process that implements the autonomous communicating functionality of an application. Software standard specification for agent based systems is provided by the FIPA in 1996 (Chris, 2005). After that number of books, research papers and thesis has been produced due to interesting area. These systems are used to provide solution for variety of problems like web search engines, warehouse management, distributed computing, mining, traffic management, tutoring systems, control systems and intrusion detection systems etc. The agent based software models used today are SWARM, StarLogo, Ascape etc.

1.7 Statement of the Problem

The traffic control problem can be further divided into two sub-domains, it can be the air traffic or road traffic control problem. We are considering the road traffic control problem and in more technical term, it is "Automated road traffic control problem". As we know that software engineering is the term used for applying systematic and quantifiable approach applied to architecture, design, development and maintenance/testing of software systems and the "automated road traffic control system" is a software system so there can be many software engineering issues regarding its architecture, design , development and maintenance/ testing.

There is a need of intelligent architecture for traffic management that solves the traffic management problem with ability to decrease average waiting time; dynamic time allocation of each side and signal according to number of vehicles; and handling of rescue vehicle.

1.8 Research Question

1) What are the problems in traffic control system?

2) How to manage rescue vehicle during the traffic congestion?

3) How to intelligently manage the traffic congestion via traffic signal?

1.9 Motivation

The problem in human administrated traffic management system is poor response time of human administrator due to the large number of vehicles or roads while on the other hand in automatic system constant time allocation to each side creates traffic congestion, wastage of time and energy. The other problem in both systems is lack of entertaining the rescue or emergency vehicles. Our research is different from the previous work in a number of ways. Previous architecture for automated traffic control and management handles the emergency or rescue vehicles by allotting separate lane that is not feasible solution in third world countries. Our proposed architecture detects the rescue vehicles and sends this information to next signals in advance so that vehicle can be able to pass each signal smoothly. The distinction from the previous work is use of dynamic time allocation for each side of the intersection. The other difference is the allocation of total signal cycle time (total time for all four sides). The proposed research will provide the architecture that provides optimized solution for traffic management by considering all relevant factors and constraints. Time distribution to vehicles in such a manner that the average vehicle's waiting time will be reduced. The proposed architecture will be simulated and results will be verified.

1.10 Application of the Research

Number of vehicles has been increased in cities due to the economy growth. Importance of smooth traffic management system is also realized to overcome crowding and accidents. The required traffic management system not only manages the traffic smoothly but also detects the emergency and over-speed vehicles with automatic mechanism for counting the number of cars. In this regard, the proposed research introduces the intelligent architecture that provides an efficient solution for traffic management. The proposed architecture considers all the relevant factors and constraints that can affect the traffic and distribute the time to each vehicle in such a way that the average vehicle's waiting time should be reduced. We have also provided the simulation of the proposed architecture to represent its strength.

Following organizations and industries can get benefit from Autonomous Traffic Management architecture.

- Capital Development Authority
- City Traffic Police
- City Police
- Intelligence Agencies

• Organization of the Thesis

In Chapter 1, we have explained the introduction and background about the traffic signals. The chapter also discusses the brief introduction of signals, their history and types. At the end of the chapter, statement of the problem, purpose and application of the research are presented. Chapter 2 is dedicated for the detailed description of state of the art that is related with traffic management and multi-agent systems.

In Chapter 3, there is discussion about the proposed Intelligent Agent Architecture for Traffic Control. Chapter 4 is dedicated for the design of proposed Intelligent Agent Architecture for Traffic Management. The chapter includes the UML diagrams. Chapter 5 discusses the dynamic time allocation through ABAC architecture that reveals the analysis and comparative results of our proposed architecture over the fixed traffic signal control. Chapter 6 presents the simulation and experimental results to show the effectiveness of our proposed architecture. The last section of the Chapter 6 is dedicated for the conclusion and future directions.

Chapter 2: Related Work

In chapter 1, we have described some of the previous traffic management systems and agent oriented paradigm. This chapter is dedicated to discuss the state of the art related with existing traffic management architectures. At the end of this chapter, the statement of the problem, motivation and application of the proposed research are presented.

2.1 State of the Art in Traffic Management

Much research has been done in the context of autonomic computing and agent oriented methodologies. In the context of traffic management, a number of systems have been developed that adopt the autonomous and agent technology. For example, (Tavladakis, et al., 1999) presented a technique that controls the traffic in a distributed fashion. Their traffic management is performed by placing the controllers at intersection points only. These controllers collaborate by exchanging data among themselves. The system uses GIS principles and acts autonomously in the sense that it can control the traffic for isolated nodes and towns. Nevertheless, these remote controllers run under the direct supervision of human being at the command centre.

(*Lee, et al., 2008*) proposed a Traffic Parameter Measure Algorithm (TPMA) to derive traffic parameters from video image sequences. The algorithm analyzes images and observes vehicle movement to extract the traffic parameters such as vehicle quantity, speed, etc. The proposed algorithm does not require any special image processing hardware and can work on a simple PC. TPMA analyzes frames based on different parameters such as brightness, noise, and many others in combination. The algorithm was validated via empirical experiments and found that the manual and proposed algorithm converges to the same count of vehicles.

(*Alagar, et al., 2003*) discussed a model for autonomous traffic control system that uses a decentralized approach where a separate controller for each lane is used to dispatch the traffic information to an arbiter. The arbiter has to allocate time slots such that the road conditions remain collision free. There are twelve such sub-systems and one arbiter. The proposed technique is described by visual and formal description in UML using class, state chart and collaboration diagram for road traffic system.

(*Sam, et al., 2001*) discussed the vehicle routing problems (VRP) and suggested the solution through agent architecture. Their proposed architecture consisted of distributed omni search strategy that solved the variants of the VRP problems without changing the actual system. Moreover, this architecture is featured by the ability of solving complex variants without rewriting the algorithms. The presented solution taps into well-known AI algorithms like genetic algorithms, simulated annealing and neighborhood search strategies.

(*Hewage, et al., 2004*) optimized the timing of traffic signal through a special purpose simulation tool. This tool has the capability to optimize the timing of signal light at single as well as multiple junctions. The tool requires less resources and the traffic administrator who even do not know about the simulation can draw road network. He simply positions the road network through icons and can draw junctions with entering the actual traffic demand. The tool provides city network and results in visual form. Administrator can understand and analyze outputs through statistical data that are based on simulation. However it is important that actual results can only be obtained if input data is correct. The simulation tool is developed in Simphony 1.05.

(*Liu, et al., 2007*) used multi-agent architecture to design an intelligent transport system (ITS), which provides information about the road emergency or other unexpected event to drivers which help to make correct decisions. ITS mainly consists of information processing application and road condition transferring module. Information processing application module is responsible to collect analyze traffic data, while road condition transferring module is to make reaction according to the road condition information. There are two agents in ITS, which are Manger agent and Vehicle agent. Manager agent creates or deletes the Vehicle agent, collect and distribute messages. The Vehicle agent explore GIS and GPS, produces maintenance graph, select best path, check emergency areas and send this information to Manager agent. For communication they used different wireless technologies such as ad hoc network, GPS, GIS, WIMAX and WiFi.

(*Albagul, et al., 2006*) proposed an algorithm and simulation for intelligent traffic signal which detects the vehicles and provides the green signal time accordingly. Authors have removed the traffic congestion by providing the appropriate time to signal through a mathematical function. The proposed algorithm for traffic control is simulated in MATLAB and the countdown timer is also developed in Lab VIEW software.

(Huang, 2006) provided a model for the traffic light control using the state chart diagram which provides the visual formalism of the system. Author performed the structure analysis of the state chart and used to show the 2, 6 and 8 phase traffic lights; and illustrated the concurrency, synchronization and causality of the system. The research also introduced a new methodology for modeling the system and named as concurrent state graph. The introduced methodology represents the concurrent states in complex state charts.

(Shamshirband, et al., 2008) introduced a multi-agent and Weighted Strategy Sharing (WSS) technique for cooperative learning in traffic management system. According to WSS, each agent measures the effectiveness of other agent in the system, assigns weight and learns from other accordingly. They also discussed the criteria by which expertness of the agents is calculated. According to author, the proposed technique is found to be effective after testing it for three traffic light system and compared it with non-cooperative agents. The said technique however exhibit sever problems like learning time too high for large number of intersections and very simple simulation that is not reflecting the real situation.

The research in *(Sadek, et al., 2006)* proposes an intelligent agent based solution for dynamic traffic routing where the agent is created in simulated highway model and learns by interaction of simulated model. The proposed model is deployed in real situation after agent reaches at satisfactory performance level by learning. After deploying in real situation it learns from its environment and updated itself. The traffic is simulated with simple highway and two alternative routes. The simulation results are effective as the learner agent route the traffic dynamically to maximize overflow. Authors also claimed that the proposed approach can be applied to other similar real world problems.

(Ranjanni, et al., 2011) discussed and proposed a model based on UML for adaptive traffic control system. The model uses detectors to control the traffic on highway. Traffic flow is controlled by coordinating all the signals of the city and exchanging their information with each other. This model is intelligent and adaptive as it changes the duration of the signals with the fluctuation of traffic. The traffic moves and stops according to signal's plans that are generated according to their dependency.

The problem with human administrated traffic management system is poor response time of human administrator due to the large number of vehicles or roads. On the other hand in automatic traffic management system allocation of constant time to each side creates the traffic congestion, wastage of time and energy. The other problem in both systems is lack of entertaining the rescue or emergency vehicles. Our research is different from the previous work in a number of ways. Previous architecture for automated traffic control and management handles the emergency or rescue vehicles by allotting separate lane that is not feasible solution in third world countries. Our proposed architecture detects the rescue vehicles and sends this information to next signals in advance so that vehicle can be able to pass each signal smoothly. The distinction from the previous works are use of dynamic time allocation for each side and total signal cycle time (total time for all four sides) of the intersection. The proposed research will provide the architecture that provides optimized solution for traffic management by considering all relevant factors and constraints. Time distribution to vehicles in such a manner that the average vehicle's waiting time will be reduced. Currently the proposed traffic management architecture is tested through a simulation.

Chapter 3: Intelligent Agent Oriented Architecture For Traffic Management

In this chapter we are presenting the architecture for traffic management that will handle the traffic problems intelligently and without the involvement of any human. The proposed architecture is named as Agent-Based Autonomous Controllers (ABAC). We are also discussing the algorithms that are involved in ABAC architecture.

3.1 The ABAC Traffic Management Architecture:

Traffic control problem can be solved by using the autonomous agents. In this paper, we introduce the Agent-Based Autonomous Controllers architecture for the traffic management. In this architecture, each component of traffic control architecture is represented by an autonomous agent. These agents are responsible to take decisions and actions on their own according to predefined desires and requirements. The abstract view of the proposed ABAC architecture is shown in figure 2 which consists of supporting/ environmental components, knowledge base and observer agent. Supporting components include cameras, sound sensors and actuators; Knowledge Base is the data repository; while the observer agent consists of three sub-agents, i.e. Analyzer agent, Decision-Making agent, and the Learner agent.

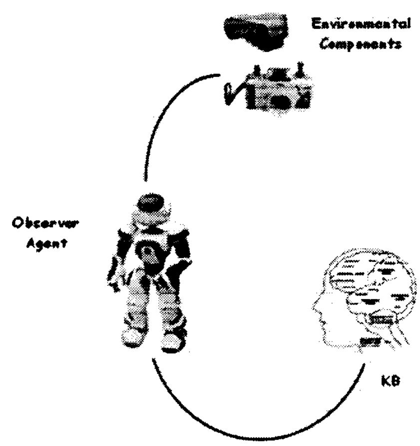

Figure 4. Abstract view of the ABAC Architecture

The ABAC architecture collects traffic data using sensors, cameras or sonic sensors. The collected information is sent to the Analyzer agent who processes the image and counts the number of vehicles on each side of the intersection. The vehicle count information passes to the

Decision Making (DM) agent for the allocation of suitable time interval. The DM agent analyzes the counted vehicles of each side and allocates threshold for a signal. The threshold gets updated every hour after analyzing the traffic data that will be retrieved from the knowledge base. After the allocation of threshold, the time of each signal will be allocated according to the traffic on particular side. The steps for time allocation to each side will be repeated after the signal threshold. In case of receiving an alert about a rescue vehicle from the sensor, the DM agent stores the current state of signals into the Knowledge Base, closes the opened traffic signal after a while and opens the signal from which the emergency vehicle is approaching. After passing the rescue vehicle, the previous state of the signals and opening sequence will be retrieved. After restoring the previous sequence, the rescue vehicle information will be sent to next signal. The steps of the ABAC architecture are discussed above at an abstract level. The detailed view of the ABAC architecture is shown in Fig. 3 and discussed in the coming section.

3.1.1 Environmental Components

i) ***Sensor:*** The sensor in ABAC architecture is used for monitoring which can be a camera and sonic detector. The camera at each side is used to capture the image of the incoming traffic while sound sensor is required to detect the emergency, rescue or other specific alarms. The captured information of image or sound is communicated to the Analyzer agent for further analysis and processing.

The ABAC adopts the following workflow: System takes input in the form of images from cameras and sound alert from the sound detector.

Casel: Image Captured from Camera:

The image on each side is to be taken 10 seconds before the completion of signal cycle time or threshold. After taking the image, it is processed by counting the number of vehicles on each side. The time to each side is given on the basis of traffic flow at each side. After completing the signal cycle, decision will be made whether to continue the current signal or switch to the next one. The waiting time of each vehicle depends upon the traffic flow of all four sides and signal status. Rest of the detail about it is discussed in coming section.

Case2: Sound detected by Sonic Sensors:

When the rescue, emergency or any other special sound is detected through the sonic sensor, the sound detection algorithm (Casey, 2002). starts its working and matches with stored sounds. The rescue and other vehicle sounds are already stored in the knowledge base with proper indexing. After the successful match, this information is sent to the Observer agent to take further action and reject in case of no match. After receiving the successful search, the Observer agent stops all other activities to provide service to rescue vehicles in a proper and intelligent manner which is described in coming section.

ii) ***Actuators:*** The actuators switched the traffic signals to yellow or green or red. It works as an execution component and controlled by the Decision Making agent.

3.1.2 Knowledge Base

It is a repository for different types of information such as threshold of each side of the signal, distance of other nearest signals from all its four sides, current opening sequence of a signal and rescue alarms. All the agents use the knowledge base to retrieve and store the information for further processing.

3.1.3 Observer Agent

The main function of observer agent is to regulate the traffic flow. It not only synchronizes with its internal agents but also synchronize with the environmental components. The observer agent allocates the proper amount of time and provides the collision free passage. It consists of three sub-agents, Analyzer, Decision-Making and Learner agent. These sub-agents of the Observer Agent are described as under:

Analyzer Agent: It is responsible for taking input from the sensor; and decision about the dependent as well as independent signal on the basis of distance to next signal; and process accordingly. The input from sensor consists of image or sound with direction. In case of image, it will be preprocessed by removing the noise and irrelevant data.

The analyzer agent counts the number of vehicles on each side of the signal by using the Background Subtraction Algorithm (Cheung, et al., 2004, 2005). This algorithm consists of four steps which are preprocessing, background modeling, foreground detection and finally data validation. The processing step take the image and removes noise by spatial, temporal smoothing and morphological processing. The background modeling step is used to describe the statistical status of the background. The foreground detection step recognizes those pixels of the image that were not identified in background modeling step. The output (candidate foreground mask) of this step is in a binary form. At the end, data validation is performed by eliminating the pixels that are not a part of moving objects and generates foreground mask. Number of vehicles will be counted by differentiating the two frames. Number of vehicles counted on each side of the signal is sent to the decision-making agent. If the incoming data from the sensor has rescue alarm/ sound then Analyzer agent stops all other processing and sends this information to the DM agent for quick response. The emergency and rescue vehicles can also be detected from pre-processed image. However, we are using sound sensor approach, as the rescue vehicle may be far away or hidden behind other vehicles in image.

Decision Making Agent: The DM agent receives the input from the sensor or Analyzer agent. In case of sound input from the sonic sensor, the DM agent will turn the opened signal to red after a while (10 secs); store the current sequence into the Knowledge Base; and turn the signal to green from where emergency vehicle is coming. After its passage, the DM agent will close the current signal after a while (10 secs) and signals opened according to the actual sequence that is stored in the Knowledge Base. DM agent will send this information to the Observer agent of the next signal, which organizes the situation to handle incoming emergency vehicle. The observer agent of the next signal will calculate the arrival time by using the distance formula.

When DM agent receives input from the Analyzer agent; it calculates the number of rows by dividing the total number of vehicles to number of lanes. In case of independent signal, assigned time will be calculated by adding the exit time of first and next rows. The exit time for first row is calculated by using the formula 3600/10000*Distance, where 3600/10000 represents the time to cover a meter when vehicle's average speed is

10 km/hour, because vehicle initiates with 0 speed and finally reaches up to 20 km/hour. Exit time for next rows will be calculated for 2 meters which is the distance of vehicle from its successor row. All four sides will be assigned time by using the same strategy. After the allocation of time, the signals will be opened from least to maximum time and continue for one hour. The alert time will also be included in total signal time. Threshold will be set automatically by taking the first time image of each side (section V) and will be updated after each an hour. In any case, when assigned time of the signal is greater than its threshold then DM agent uses the threshold rather than assigned time.

In case of dependent signal, first sum up the number of vehicles of dependent signal, 70% from the straight signal and 30% from the tilt signal. After summation, the assigned time will be calculated by adopting the same strategy as mentioned for independent signal.

Learner Agent: The learner agent takes the data from the knowledge base, identifies useful patterns and on the basis of these patterns improves the knowledge base. The improved knowledge will be used by the analyzer and DM agent to take future decisions more intelligently. Learner agent improves the accuracy of signal time for each side, signal cycle time, dependent signal reachability time and the arrival of emergency vehicle to the next nearest signal. Functionality of the learner agent is further divided into three sub-modules which are used to optimize the signal time, emergency vehicle time and signal's cycle time. The optimization will be performed by using the Bayesian algorithm that takes previous history and improves the behavior.

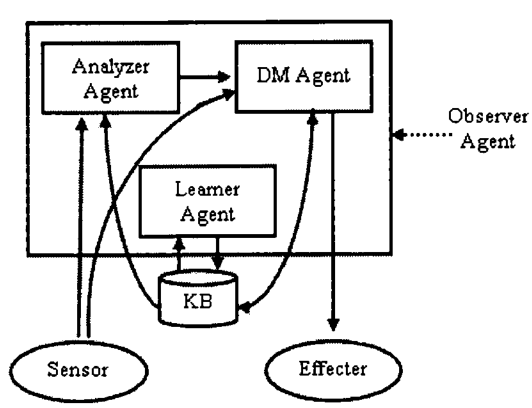

Figure 5. Architecture of the ABAC System

3.2 ABAC Algorithms

The ABAC architecture operates on top of three main algorithms: The FullRoundThreshold algorithm, the CountVehicles algorithm, and the SoundAlert algorithm. More details about these algorithms are to follow.

3.2.1 FullRoundThreshold Algorithm

The FullRoundThreshold algorithm (Fig. 4) is used to set the threshold for the maximum time needed to complete processing the traffic flow at the four sides of an intersection. This algorithm is invoked on regular basis in order to update the threshold (t) and keep it adaptive to the intensity of the traffic flow. We update threshold using the exponential moving average technique, but any smoothing technique can be used instead. Here is a sketch for how this algorithm works for each signal:

1. Obtain a time series of the signal time from the knowledge base which have been collected over the last one hour time window with the help of the CountVehicle algorithm (explained below).
2. Calculate signal's average by dividing sum of total time obtained from (1) over the number of readings.
3. The calculated average time will be set as threshold such that it respects the minimum and maximum limits of [40, 200] range.

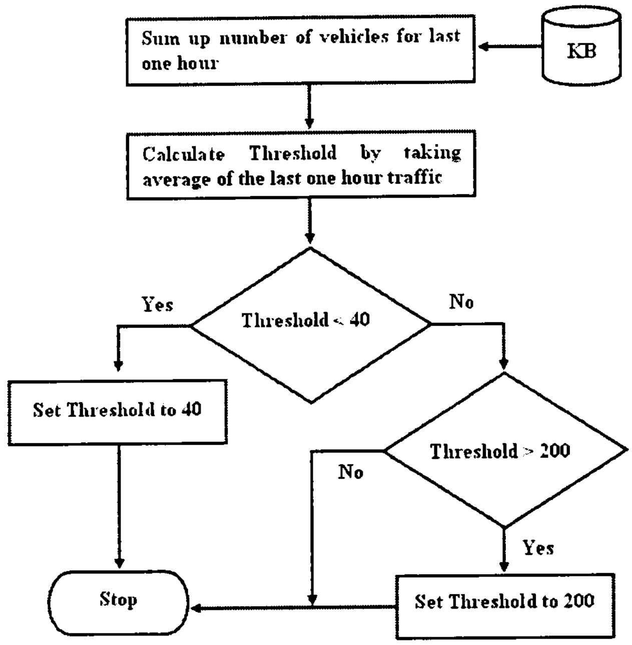

Figure 6. FullRoundThreshold Algorithm

3.2.2 CountVehicles Algorithm

The goal of this algorithm (Fig. 5) is to populate the knowledge base with a time series of vehicle count at each signal on regular basis. It uses image processing technique to discern vehicles and count them. Briefly, this algorithm carries out the following tasks:

1. Takes image of vehicles on each side of signal.
2. Process each image by counting the vehicles.
   a. Capture the two consecutive image frames.
   b. Store the relevant data of image to an array.
   c. Take difference of two consecutive array elements.
   d. By taking the difference, count the number of vehicles on each side.
3. Number of vehicles of four sides is compared with each other and threshold.
4. Store number of vehicles on each side, total vehicles of a signal along with their assigned time.
5. On the basis of comparison and previous signal information (number of vehicles and time to reach), the time is allocated to each side of the signal as:

a. Take summation of vehicles of each side and vehicles of previous signal that reach within a threshold time.
b. Take threshold of a signal from the knowledge base.
c. If the previous signal's vehicles take less time to reach than threshold value then
    i. Calculate factor of these sides by adding the vehicles of current signal with previous and subtract the time to reach from the previous signal.
    ii. Add calculated factor into current vehicles.
d. Calculate the time for each side by dividing the number of vehicles of that side over total signal's vehicles and multiplying it with signal threshold.
    i. If the system is already not following any sequence, then open the signal having minimum time.
    ii. Store the current sequence and total number of vehicles with time into the knowledge base.
    iii. Send the information (number of vehicles, time to reach) of each side to its next connecting side if exists.

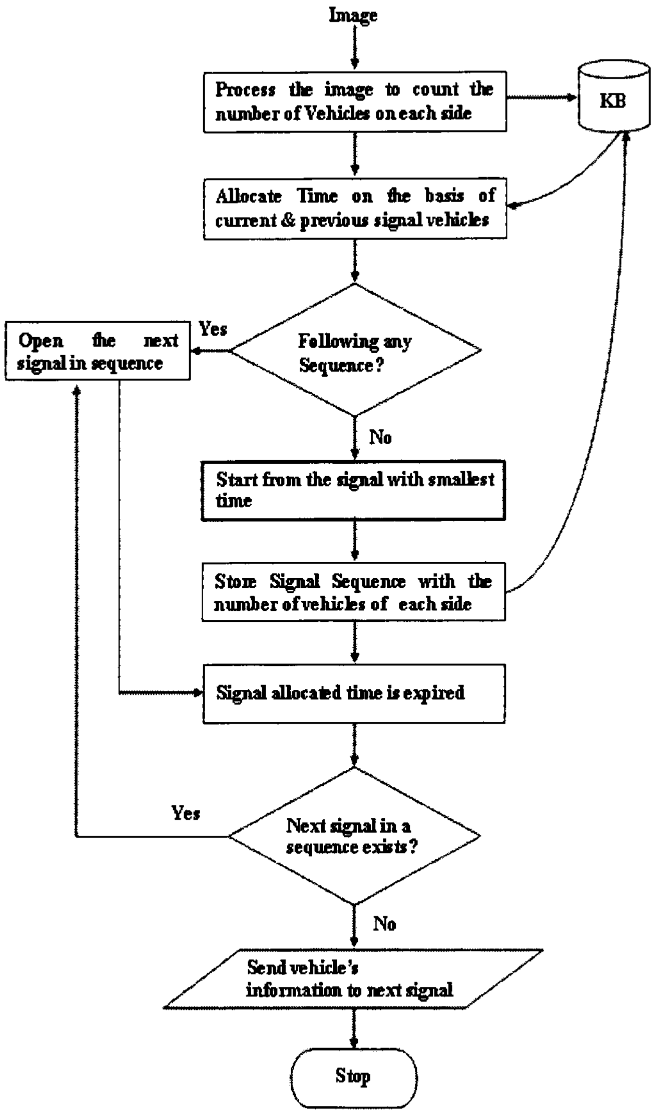

Figure 7. CountVehicle Algorithm

3.2.3 SoundAlert Algorithm

This algorithm (Fig. 6) sets of when the sensor intercepts the siren of a rescue vehicle. Briefly, the algorithm goes through the following steps:

1. Take the distance and coordinates of the emergency vehicle.
2. Change the green signal (from the side where emergency vehicle is approaching) to yellow for 10 seconds and close the current opened signal.
3. Store the remaining time of opened signal into the knowledge base.

4. Open the signal from where emergency vehicle is approaching and remained opened for 10 seconds more than its allocated time.
5. Send the information about the emergency vehicle to the next signal.
6. Execute the stopped signal for remaining time and continue to previous sequence.

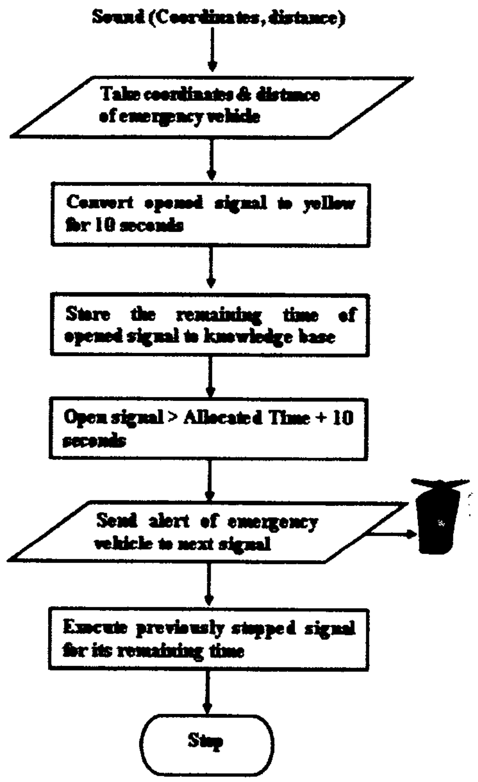

Figure 8. SoundAlert Algorithm

Chapter 4: System Design

In this chapter we are presenting the architecture for traffic management through UML diagrams. These diagrams include Use Case, State Chart, Class and Interaction diagrams. The Interaction Diagram is further divided into Sequence and Collaboration diagram,

4.1 ABAC Architecture:

We can define the system architecture as the conceptual model which is used to state the organization, behavior and views of a system. The system architecture helps and describes the layout of the system components, characteristics of the components, their relationship and an overall plan to produce the system.

In order to get functional and nonfunctional requirements of ABAC architecture we have model it through UML diagrams. These diagrams include uses case, class, state chart, sequence and collaboration diagrams. These UML diagrams are described as follows:

4.2 Use Case Diagram

Use case diagram provides interaction between actors and a system. In ABAC architecture there are three actors which are camera, sound detector and actuator.

4.2.1 Use Case Diagram for Sensor (Camera)

The Use Case Diagram for Sensor (Camera) is described in Figure 9 that represents three functions of the camera which are set the camera, take the image of current traffic and send this information to the actuator for further action.

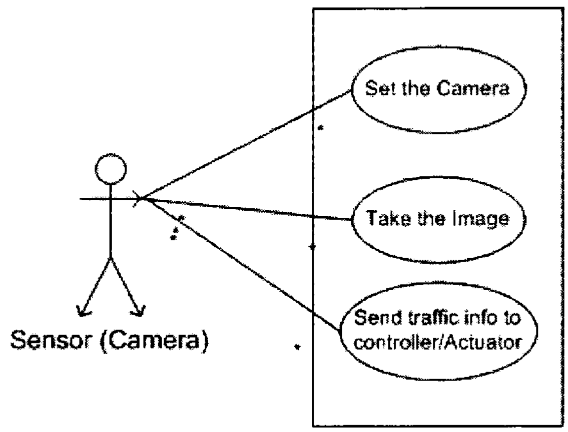

Figure 9. Use case diagram for Sensor (Camera)

4.2.2 Use Case Diagram for Sensor (Sound Detector)

The Use Case Diagram for Sound Detector is described in Figure 10 that represents three functions which are set the sound detector, Detection of rescue alarm and send the information about sound to the actuator for further action.

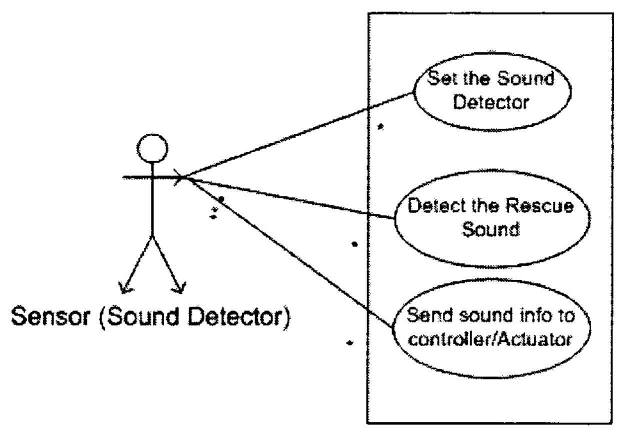

Figure 10. Use case diagram for Sensor (Sound Detector)

4.2.3 Use Case Diagram for Actuator

The Use Case Diagram for Actuator is described in Figure 11 that shows the functions performed by the actuator. The actuator performs its function after receiving the information of traffic or sound from sensors.

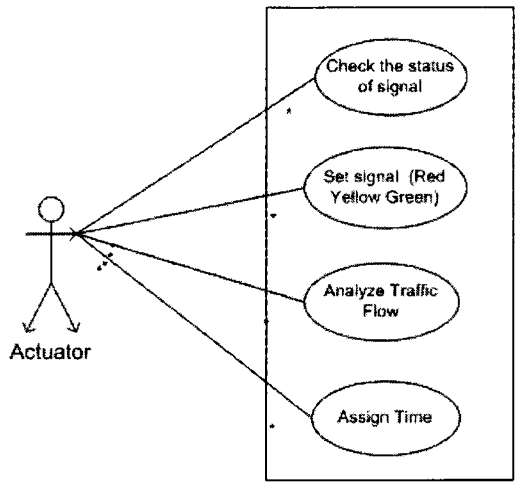

Figure 11. Use case diagram for Actuator

4.3 Class Diagram

Class diagram represents a static view of system where the classes are shown with their attributes, methods and relationship. The following class diagram shows the abstract as well as detailed modeling.

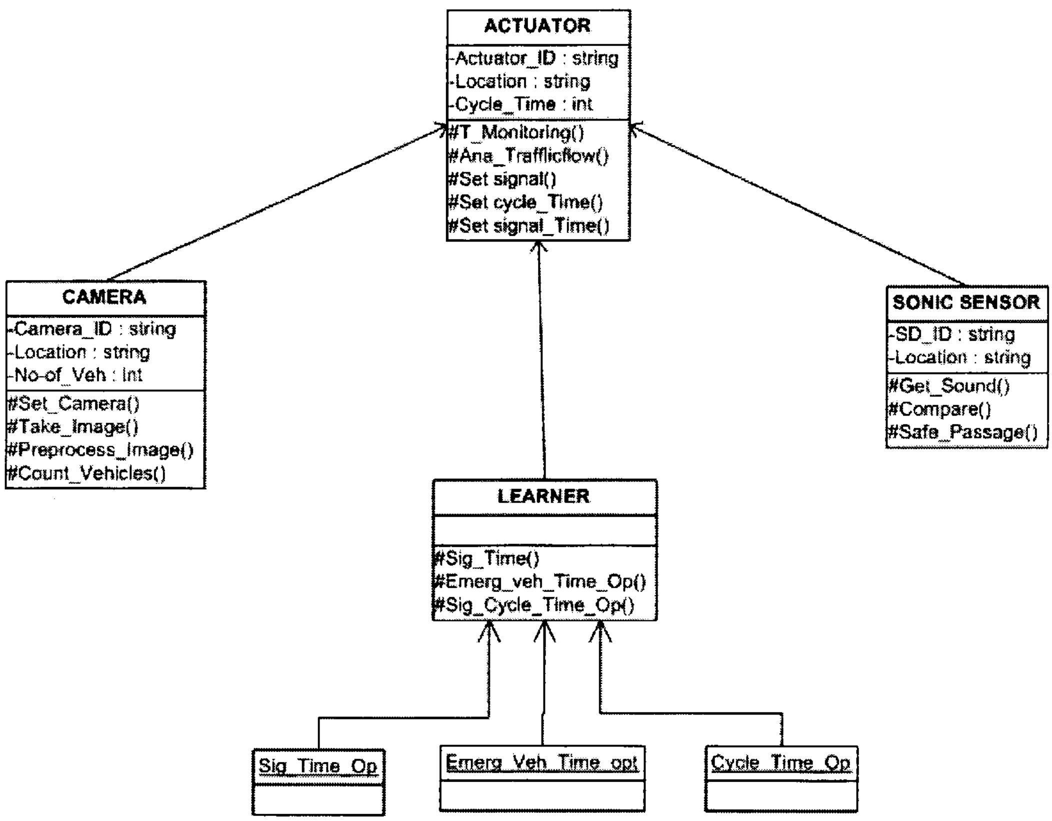

Figure 12. Class diagram for ABAC

The above class diagram of ABAC describes that there is one main class i.e. Actuator. It is further divided into three sub-classes which are Camera, Sonic Sensor and Learner class. The camera class represents the location and vehicles count while the sonic sensor class represents the location with sound detection information. The Learner class is responsible for the optimization of signal time, emergency vehicle time and cycle time.

4.4 State Chart Diagram

In UML, State chart diagram represents the sequence of object states during its life time. These different states occur due to actions and responses. State chart of the signal states which are wait, ready to go and go as shown in figure 13.

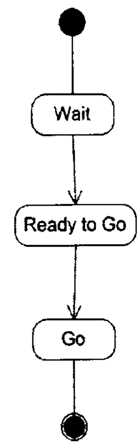

Figure 13. State Chart diagram for Signal

State chart of the signal lights is shown in figure 14. The states of the signal lights are green, yellow and red. These states are changed according to some rules. Green light remains either green or can be switched to yellow light, Yellow light can be switched either to green or red and finally red light remains red or switched to yellow.

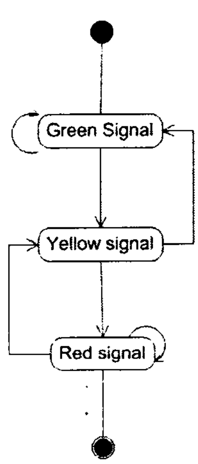

Figure 14. State Chart diagram for Signal Lights

4.5 Interaction Diagram

Interaction diagram is the dynamic diagram and depicts the use case behavior to complete certain activity or task. There are two types of interaction diagram, sequence and collaboration.

4.5.1 Sequence Diagram

Sequence diagram is used to represent the object and messages between these objects. The focus of the sequence diagram is on the sequence of the messages. The sequence diagram of ABAC is shown in figure 15.

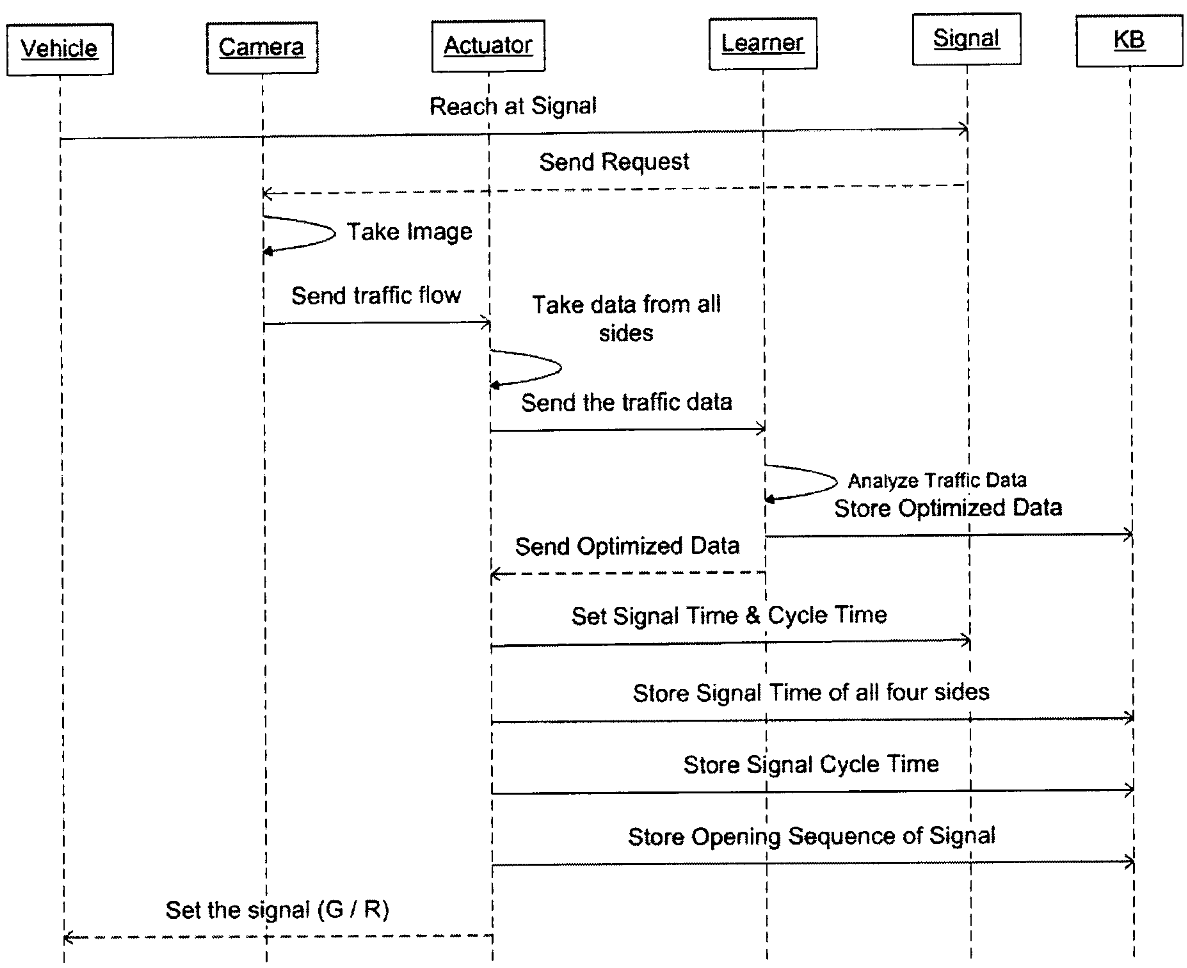

Figure 15. Sequence diagram for ABAC

4.5.2 Collaboration Diagram

Collaboration diagram focuses on the relationship between the objects. The relationships are presented in visualized form.

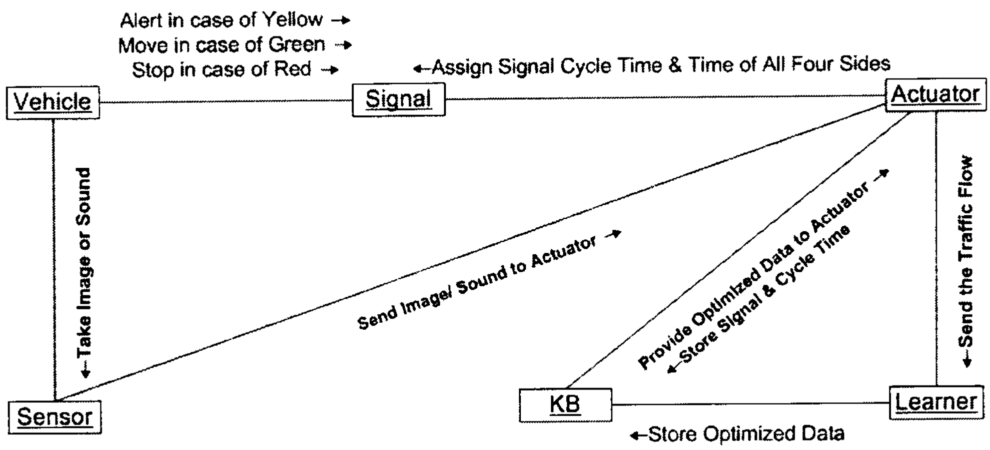

Figure 16. Collaboration diagram of ABAC

Chapter 5: Dynamic Time Allocation through ABAC Architecture

This chapter presents the dynamic time allocation through ABAC architecture where we have categorized the traffic signals into two types, i.e. Independent and dependent traffic signals. Here dependency means traffic of one signal is dependent on other signal's traffic.

5.1 Classification of Signals

Traffic congestion is the common problem in big cities of modern and third world countries. Developed countries have already paid keen attention to traffic congestion problem and are solving it by adopting intelligent strategies. Pakistan is also facing the same situation and still implementing the static traffic management architecture which causes wastage of time and energy, insecure drive and delay in rescue services. Serious attention is required to resolve this issue by adopting intelligent and autonomous traffic management architecture. Here the traffic signals are divided into two types according to their relationship, i.e. independent and dependent signals.

5.1.1 Independent Traffic Signals

Following formulae are used to calculate the different parameters for the allocation of signal time:

Assigned Time = Exit Time (for 1st row) + 0.72 (Rows - 1)

Exit time (for 1st row) = 3600/10000 * Distance

Exit time (for next rows) = (3600/10000 * 2) (Rows -1)

= (0.72) (Rows -1)

Number of Rows = Round (No. of vehicles / No. of Lanes)

Where

Exit Time: Time to cross the signal

Distance: Signal Width

If (Assigned Time > Threshold) then

Assigned Time = Threshold

In calculation of vehicle exit time for first row the 3600/10000 represents the time to cover one meter when average speed is 10 km/hour, because generally vehicle starts with 0 speed and when crossed the signal speed reaches up to 20 km/hour. Exit time for next rows will be calculated for 2 meters which is the distance of vehicle from its successor row.

Suppose threshold for individual signal is 50 secs and numbers of vehicles at each side of the signal are 20, 9, 4 and 17. These are represented through variables V1, V2, V3 and V4 respectively. Here we are considering a road with 2 lanes with 10 meter of distance. If the assigned time for one side of the signal is less than 50 seconds then the assigned time is used for further processing, however if the assigned time is greater than 50 seconds (Threshold) will be used as a signal time.

Signal 1

Assigned Time = Exit Time (for first row) +0.72 (Rows-1))

T1 = 3.6 + 0.72 (9)

= 10.08 sec

If (10.08 > 50) then

T1 = 50

T1 will remain 10.08 sec.

Signal 2

Assigned Time = Exit Time (for first row) + 0.72 (4)

T2 = 3.6 + 2.88

= 6.48 sec

Signal 3

Assigned Time = Exit Time (for first row) + 0.72 (3)

T3 = 3.6 + 2.16

= 5.76 sec

Signal 4

Assigned Time = Exit Time (for first row) + 0.72 (8)

T4 = 3.6 + 5.76

= 9.36 sec

Total Signal Time (Cycle Time) = Sum (Signal Time) + Alert Time

= (10.08 + 6.48 + 5.76 + 9.36) + 20

= 51.68 sec

(Where 20 seconds is the time for alert/ yellow light time for a signal, which will have same effect on static traffic management architecture and the proposed architecture)

In above case the order of signal opening will be signal 3, 2, 4 and 1. When the time allocated to signal 3 is completed or all vehicles are being passed from that side, the DM agent will turn on the yellow light for a while (5 sec) and switch to red and then open the next signal i.e. signal 2 for above case and this process continue for next signals. After the completion of each cycle, the above process will be repeated. The average waiting time for above example will be calculated as:

Signal 3 Average Waiting Time = 0

Signal 2 Average Waiting Time = T3 / V2 = 0.64

Signal 4 Average Waiting Time = (T3 + T2) / V4 = 0.72

Signal1 Average Waiting Time = (T2+T3+T4) / V1 = 1.08

Average Waiting Time for All Vehicles = 2.44

Table 4 at the end of this paper presents the effectiveness of proposed ABAC architecture over the static traffic management architecture that is represented in Table 5, where constant and same time is allocated to each side on the basis of history and previous experiences. According to the scenario, where different number of lanes, vehicles and number of rows are considered. In table 5 the Assigned time and average waiting (Avg WT) are computed on the basis of above formulae where in table 5 the column Assigned Time is already allocated according to the static traffic management technique and column Average Waiting Time (Avg WT) is calculated by using the above formulae. To avoid confusion, we take data in such a manner that traffic signals will always be opened in 3, 2, 4, 1 sequence. The average waiting time of signal 3 in both cases is 0 as signal opening sequence starts from it. The comparison of average time in both tables represents that how much time will be saved by adopting the proposed ABAC architecture.

In table 6 each side of the signal is allocated a constant time on the basis of history and previous experiences. The previous history shows that the average number of vehicles on side 1, 2, 3 and 4 is 26, 14, 11 and 20 respectively. The constant time is allocated to each side by considering the average vehicles. In this way the time is allocated to signal 1, 2, 3 and 4 is 37, 19, 16 and 28 secs respectively. In table 8, we have taken 70 secs as the signal cycle time which is equal to the signal cycle time of table 4. The time to each signal is allocated by considering the average vehicles. In this case, the signal's cycle time is 70 secs so the time allocated to signal 1, 2, 3 and 4 is 25, 14, 12 and 19 secs respectively.

Table 1 represents the average waiting time of four scenarios and each column against these scenarios is taken from the Table 3-6. The range of signal's average waiting time in our proposed ABAC architecture (Column 1) is [1.12, 3.52] while in static traffic management architecture the range of signal's average waiting time of first scenario (Column 2) is [3.28, 19.64], second scenario (Column 3) is [2.44, 14.50] and third scenario (Column 4) is [1.78, 10.61]. The comparison of our proposed ABAC architecture with static traffic management architecture represents the efficiency of proposed architecture as revealed in Fig. 7. In all three scenarios of static traffic management architecture the best time is [1.78, 10.61]; however our proposed architecture saves more time. The other major problem with static traffic management architecture is allocation of constant time where the average waiting time of signal will increase even reducing the number of vehicles. Due to this reason if there is no vehicle at any side; the respective signal will remain opened for assigned time. While in our case the signal's assigned time will be automatically decreases with reduction in number of vehicles.

Table 1 : Comparison between ABAC and Static Traffic Management Architecture

ABAC Avg WT	Static Case 1 Avg WT	Static Case 2 Avg WT	Static Case 3 Avg WT
2.44	9.47	6.99	3.73
1.73	9.47	6.99	5.11
3.52	9.47	6.99	5.11
1.48	5.36	3.92	2.88
3.02	19.64	14.5	10.61
2.72	10.18	7.67	5.59
1.12	3.28	2.44	1.78
2.25	8.09	5.94	4.36
2.34	10.65	8.03	5.86
1.39	4.63	3.43	2.51

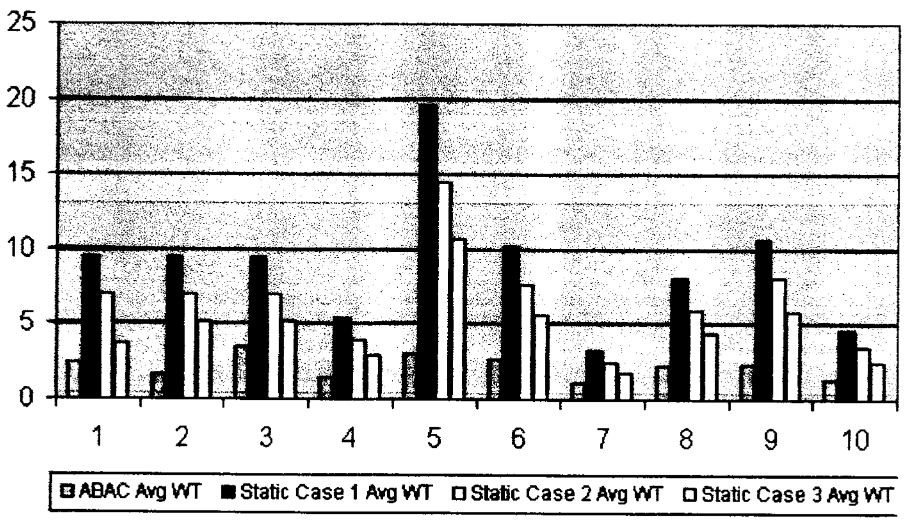

Figure 17. Average Waiting Time of ABAC Vs Static Traffic Management Architecture

5.1.2 Mutual-Dependent Traffic Signals

In the above section we have discussed the simple and independent traffic signal scenario. Here we are discussing the more complex form of traffic signals i.e. dependent signals. Two signals which are connected and close within the range of 50 meters are said to be dependent traffic mutual signals. In case of mutual dependent traffic signals, the traffic flow of one signal affect the other signal and vice versa as shown in Fig. 8. There are 7 roads and each road has two sub-roads i.e. incoming and outgoing in below figure. The assigned time of the signal 'C' depends on the traffic of vehicles that are waiting on Signal 'C', outgoing traffic of signal 'F' and 'G'. Similar is the case for signal G's assigned time that depends on vehicles waiting on E, A & D.

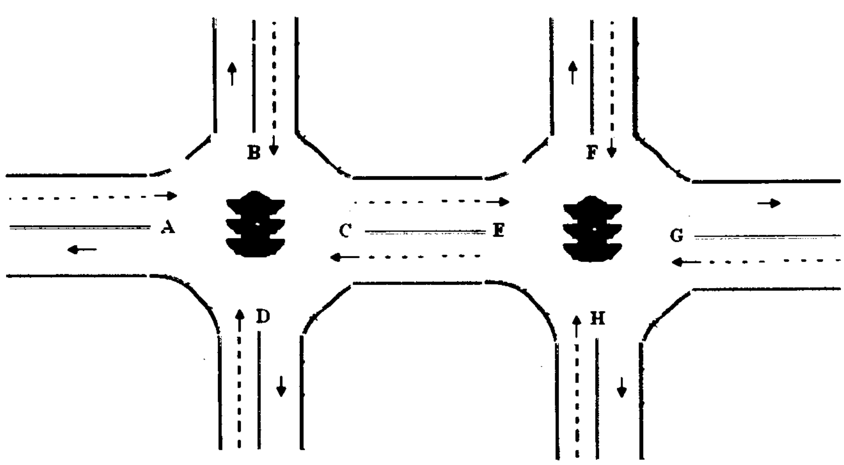

Figure 18. Mutual-Dependent Traffic Signal Scenario

Suppose we have mutual dependent traffic signal as shown in above figure, where number of vehicles at signal A, B, C and D are 13, 20, 10 and 11 and number of vehicles at signal E, F, G and H are 8, 21, 27 and 13 respectively. The above mutual-dependent traffic signals can be represented through graphs as shown in Fig. 9, where each signal is denoted through a node and the incoming edge represents traffic or number of vehicles moving towards that node. The signals of Fig. 8 are divided into two types as shown in Fig.9, i.e. independent (A, B, G and H) - only depending on their own traffic, dependent (D, F) - F depends its own as well as traffic at E and mutually dependent nodes (C, E).

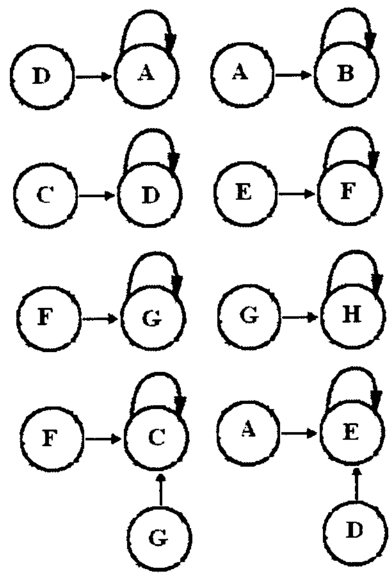

Figure 19. Graph representation of Mutual-Dependent Traffic Signal

Figure 19 can easily be understandable through an adjacency list. In graph theory, an adjacency list is a way to represent the nodes with their connectivity to other nodes. First row of the Table 2 represents the individual signal (node) of above scenario while row 2 shows the signals on which their assigned or calculated time is depending.

Table 2: Signals with Associated Dependency

Signal	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Dependent Signal	A D	B A	C F G	D C	E A D	F E	G F	H G

As the signal A, B, G, and H are independent signals that's why calculated according to above independent signal scenario. However, remaining signals which are dependent will be assigned time by using the following formulae:

Assigned Time of C = No of vehicles (C + 0.3F + 0.7G)

Assigned Time of D = No of vehicles (D + 0.3C)

Assigned Time of F = No of vehicles (F + 0.3E)

Assigned Time of E = No of vehicles (E + 0.3D + 0.7A)

In above equation 0.3 and 0.7 represents the 30% and 70% of the total vehicle that will approach towards respective signal. These ratios are taken very carefully after taking various traffic flow scenarios. After experimenting, we found that 70% of the traffic moves towards straight signal and 30% of the traffic moves to tilt side (i.e. change its direction). Following table 3 represents the assigned time to each signal by using above formulae.

Table 3: Assigned Time for Mutual-Dependent Traffic Signals

Signal	Exit Time (for 1st Row)	Exit Time (for Next Rows)	Vehicles	Assigned Time
A	3.6	0.72	13	13
B	3.6	0.72	20	16
C	3.6	0.72	10	29
D	3.6	0.72	11	11
E	3.6	0.72	8	18
F	3.6	0.72	21	16
G	3.6	0.72	27	22
H	3.6	0.72	13	16

Table 4 : Results of ABAC Architecture

S#	Lanes	Signal 1				Signal 2				Signal 3				Signal 4				Signal Time	Signal Avg WT
		Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Average WT		
1	2	20	10	10.08	1.08	5	5	6.48	0.64	7	4	5.76	0.00	17	9	9.36	0.72	51.68	2.44
2	4	20	5	6.48	0.73	5	2	4.5	0.48	7	2	4.32	0.00	17	4	5.76	0.52	41.06	1.73
3	1	20	20	17.28	1.62	5	9	9.36	0.88	7	7	7.92	0.00	17	17	15.12	1.02	69.68	3.52
4	3	40	13	12.24	0.56	16	5	6.72	0.41	14	5	6.48	0.00	26	9	9.36	0.51	54.80	1.48
5	3	10	3	5.04	1.20	5	2	4.08	0.72	3	1	3.6	0.00	7	2	4.32	1.10	37.04	3.02
6	2	16	8	8.64	1.33	13	7	7.56	0.44	8	4	5.76	0.00	14	7	7.92	0.95	49.88	2.72
7	4	55	14	12.96	0.50	32	8	8.64	0.25	26	7	7.92	0.00	44	11	10.8	0.38	60.32	1.12
8	2	26	13	12.24	0.86	12	6	7.2	0.54	9	5	6.48	0.00	16	8	8.64	0.86	54.56	2.25
9	3	15	5	6.48	1.15	12	4	5.76	0.42	10	3	5.04	0.00	14	5	6.48	0.77	43.76	2.34
10	3	40	13	12.24	0.63	21	7	7.92	0.31	16	5	6.48	0.00	32	11	10.8	0.45	57.44	1.39

Table 5: Results of Conventional Static Traffic Mnagement Architecture
(Case 1: All four sides are assigned equal constatnt time, i.e. 25 secs with signal cycle time 120 secs)

S#	Lanes	Signal 1				Signal 2				Signal 3				Signal 4				Signal Time	Signal Avg WT
		Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Average WT		
1	2	20	10	25	3.75	9	5	25	2.78	7	4	25	0.00	17	9	25	2.94	120.00	9.47
2	4	20	5	25	3.75	9	2	25	2.78	7	2	25	0.00	17	4	25	2.94	120.00	9.47
3	1	20	20	25	3.75	9	9	25	2.78	7	7	25	0.00	17	17	25	2.94	120.00	9.47
4	3	40	13	25	1.88	16	5	25	1.56	14	5	25	0.00	26	9	25	1.92	120.00	5.36
5	3	10	3	25	7.50	5	2	25	5.00	3	1	25	0.00	7	2	25	7.14	120.00	19.64
6	2	16	8	25	4.69	13	7	25	1.92	8	4	25	0.00	14	7	25	3.57	120.00	10.18
7	4	55	14	25	1.36	32	8	25	0.78	26	7	25	0.00	44	11	25	1.14	120.00	3.28
8	2	26	13	25	2.88	12	6	25	2.08	9	5	25	0.00	16	8	25	3.13	120.00	8.09
9	3	15	5	25	5.00	12	4	25	2.08	10	3	25	0.00	14	5	25	3.57	120.00	10.65
10	3	40	13	25	1.88	21	7	25	1.19	16	5	25	0.00	32	11	25	1.56	120.00	4.63

Table 6: Results of History Based Traffic Mnagement Architecture with 100 sec cycle
(Case 2: Signals are allocated constatnt time on the basis of previous histtory with signal cycle time 120 secs)

S#	Lanes	Signal 1				Signal 2				Signal 3				Signal 4				Signal Time	Signal Avg WT
		Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Average WT		
1	2	20	10	37	3.15	9	5	19	1.78	7	4	16	0.00	17	9	28	2.06	120.00	6.99
2	4	20	5	37	3.15	9	2	19	1.78	7	2	16	0.00	17	4	28	2.06	120.00	6.99
3	1	20	20	37	3.15	9	9	19	1.78	7	7	16	0.00	17	17	28	2.06	120.00	6.99
4	3	40	13	37	1.58	16	5	19	1.00	14	5	16	0.00	26	9	28	1.35	120.00	3.92
5	3	10	3	37	6.30	5	2	19	3.20	3	1	16	0.00	7	2	28	5.00	120.00	14.50
6	2	16	8	37	3.94	13	7	19	1.23	8	4	16	0.00	14	7	28	2.50	120.00	7.67
7	4	55	14	37	1.15	32	8	19	0.50	26	7	16	0.00	44	11	28	0.80	120.00	2.44
8	2	26	13	37	2.42	12	6	19	1.33	9	5	16	0.00	16	8	28	2.19	120.00	5.94
9	3	15	5	37	4.20	12	4	19	1.33	10	3	16	0.00	14	5	28	2.50	120.00	8.03
10	3	40	13	37	1.58	21	7	19	0.76	16	5	16	0.00	32	11	28	1.09	120.00	3.43

Table 7: Results of History Based Traffic Mnagement Architecture with 70 sec cycle

(Case 3: Signal s are allocated constatnt time on the basis of previous histtory with signal cycle time 70 secs)

S#	Lanes	Signal 1				Signal 2				Signal 3				Signal 4				Signal Time	Signal Avg WT
		Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Avg WT	Vehicles	No. of Rows	Assigned Time	Average WT		
1	2	52	26	25	0.87	9	5	14	1.33	7	4	12	0.00	17	9	19	1.53	90.00	3.73
2	4	20	5	25	2.25	9	2	14	1.33	7	2	12	0.00	17	4	19	1.53	90.00	5.11
3	1	20	20	25	2.25	9	9	14	1.33	7	7	12	0.00	17	17	19	1.53	90.00	5.11
4	3	40	13	25	1.13	16	5	14	0.75	14	5	12	0.00	26	9	19	1.00	90.00	2.88
5	3	10	3	25	4.50	5	2	14	2.40	3	1	12	0.00	7	2	19	3.71	90.00	10.61
6	2	16	8	25	2.81	13	7	14	0.92	8	4	12	0.00	14	7	19	1.86	90.00	5.59
7	4	55	14	25	0.82	32	8	14	0.38	26	7	12	0.00	44	11	19	0.59	90.00	1.78
8	2	26	13	25	1.73	12	6	14	1.00	9	5	12	0.00	16	8	19	1.63	90.00	4.36
9	3	15	5	25	3.00	12	4	14	1.00	10	3	12	0.00	14	5	19	1.86	90.00	5.86
10	3	40	13	25	1.13	21	7	14	0.57	16	5	12	0.00	32	11	19	0.81	90.00	2.51

Chapter 6: Implementation, Results & Conclusion

This is the final chapter of the thesis where we are elaborating the ABAC architecture through simulation. The simulation is developed in Java NetBeans IDE 7.0. Further chapter discusses the simulation of independent, dependent and interconnected traffic signals with and without rescue vehicle of different scenarios and comparative results. The results are shown in the form of graphs that are based on average waiting time of our proposed, conventional and history based time allocation.

6.1 Introduction to Traffic Simulation

Simulation is a way to analyze and produce the solution of complex, risky and costly problems (Champion, et al., 1999). The simulation provides us most effective way to handle the problem. Vehicle traffic control also falls in such type of problem and before its implementation it should be simulated. The traffic simulation will help us in analyzing, assessing and evaluating the traffic flow or congestion. Moreover it will also useful for determining the accuracy and effectiveness of the proposed traffic control architecture. Simulation can be categorized into deterministic (predictable) and stochastic. The deterministic simulation is one which produces the same result for specific input on the other hand stochastic simulation may produce different outcome for same input. The other types of simulation are discrete or continuous. In discrete simulation, the system's states are modeled and their states varies after a specific interval or event (discrete time and discrete event simulation) while in continuous simulation state of the model is determined by calculating the equations which are based on real values.

6.2 Traffic Simulation Entities

Traffic simulations are used to simulate the flow of traffic in two types of vehicle traffic network, i.e. motorway and city traffic. However, here we are simulating only the vehicle traffic control architecture which is based on city traffic flow. In city traffic the main features include the vehicles, roads, intersection points (signals) and traffic lights.

6.2.1 Simulating the Environment

The simulation environment consists of vehicles (rescue and others), connected roads and traffic signals. Each road is divided into one or more lanes and operates in one direction. Two types of environments has been modeled, one for independent traffic signal which consists of four roads connected in the form of signal and the other for dependent traffic signal which consists of two signals connecting 6 roads.

6.2.2 Simulating the Vehicles

In simulation, vehicle is described with certain parameters such as size (length and width), location (x-axis, y-axis), color and speed. The x and y-position describes the location of the vehicle and both are useful to move the vehicle in some specific direction. Number of vehicles and rescue vehicles on each side of the signal are created randomly.

6.3 Independent Traffic Signal

The type of signal which is depending only on its own traffic flow is known as the independent traffic signal. The parameters that were used in calculation are the number of Lanes, vehicles at signal (1, 2, 3 and 4) and priority if there is some rescue vehicle.

Scenario 1

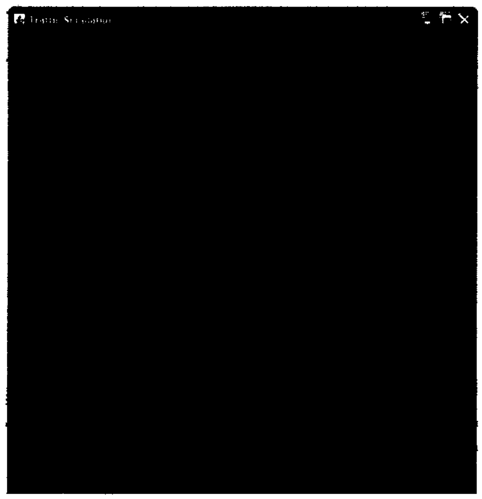

Figure 20. Independent Traffic Signal Scenario 1

In above scenario, Vehicles at signal (1, 2, 3 and 4) = 25, 20, 17, 22 and number of Lanes are 3. In other four scenarios we have taken the following number of vehicles.

Vehicles at signal (1, 2, 3 and 4) = 39, 20, 17, 32
Vehicles at signal (1, 2, 3 and 4) = 20, 10, 7, 13
Vehicles at signal (1, 2, 3 and 4) = 25, 15, 11, 13
Vehicles at signal (1, 2, 3 and 4) = 11, 8, 14, 9
Opening signal sequence adopted is 3, 2, 4, 1

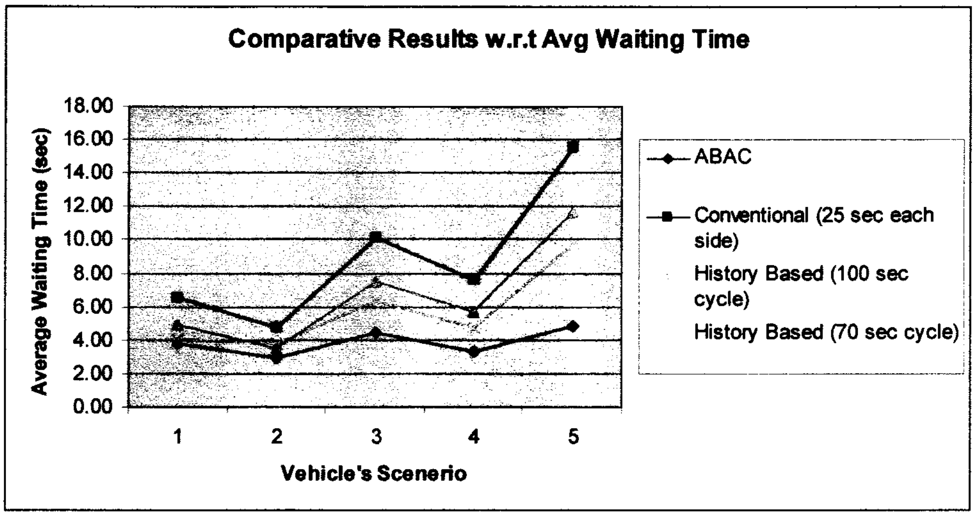

Figure 21. Graph of Independent Traffic Signal Scenario 1

Discussion

In above graph, a few case of history based 100 sec cycle time and history based 70 sec cycle time approach shows equal average waiting time than our approach. However, the overall average waiting time of other approaches is greater than ABAC. The range of the average waiting time of our approach is 2.91 to 5.44, conventional approach with 4.74 to 15.5, history based 100 sec cycle time approach with 3.51 to 11.62 and history based 70 sec cycle time approach with 3.82 to 9.77. This represent that ABAC architecture is more efficient than all other approaches.

Scenario 2

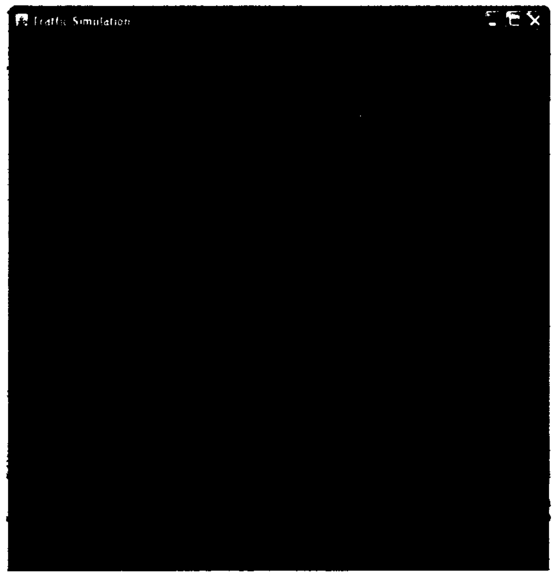

Figure 22. Independent Traffic Signal Scenario 2

In above scenario, Vehicles at signal (1, 2, 3 and 4) = 25, 9, 19, 13 and number of Lanes are 3. In other four scenarios, we have taken the following number of vehicles.

Vehicles at signal (1, 2, 3 and 4) = 16, 8, 13, 12

Vehicles at signal (1, 2, 3 and 4) = 19, 11, 16, 15

Vehicles at signal (1, 2, 3 and 4) = 10, 4, 8, 6

Vehicles at signal (1, 2, 3 and 4) = 21, 12, 19, 16

Opening signal sequence adopted is 2, 4, 3, 1

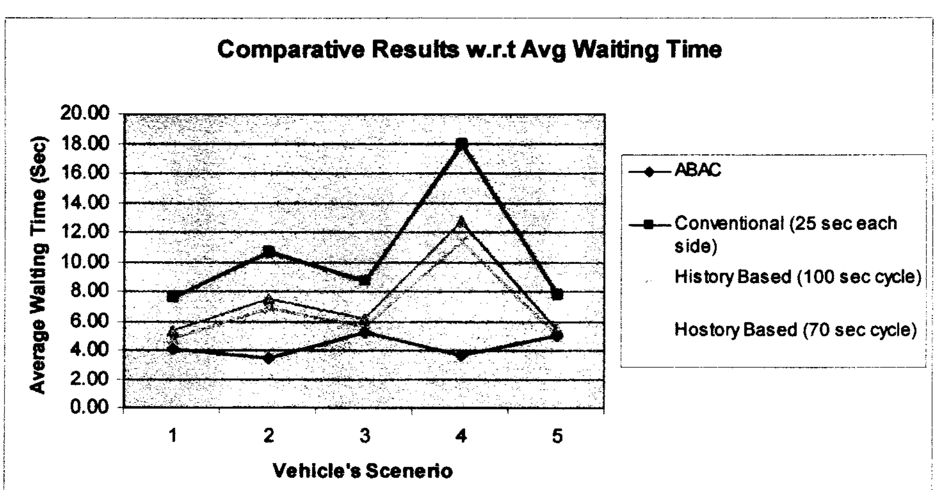

Figure 23. Graph for Independent Traffic Signal Scenario 2

Discussion

In above graph, none of the approach shows minimum or equal average waiting time than our proposed approach. The range of the average waiting time of our approach is 3.48 to 5.01, conventional approach with 7.55 to 17.92, history based 100 sec cycle time approach with 5.38 to 12.71 and history based 70 sec cycle time approach with 4.79 to 11.42. This represent that ABAC architecture in this case also depicted more efficient results than all other approaches.

Scenario 3

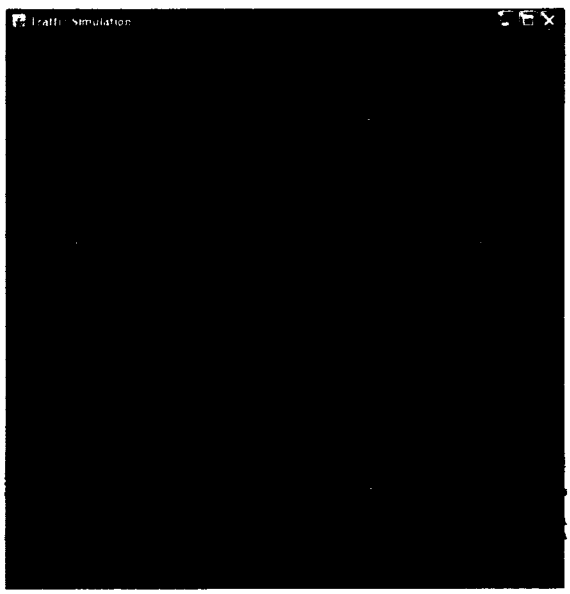

Figure 24. Independent Traffic Signal Scenario 3

In above scenario, Vehicles at signal (1, 2, 3 and 4) = 7, 11, 15, 22 and number of Lanes are 3. In other four scenarios we have taken the following number of vehicles.

Vehicles at signal (1, 2, 3 and 4) = 6, 8, 10, 12
Vehicles at signal (1, 2, 3 and 4) = 3, 5, 7, 9
Vehicles at signal (1, 2, 3 and 4) = 14, 17, 20, 25
Vehicles at signal (1, 2, 3 and 4) = 11, 18, 22, 25
Opening signal sequence adopted is 1, 2, 3, 4

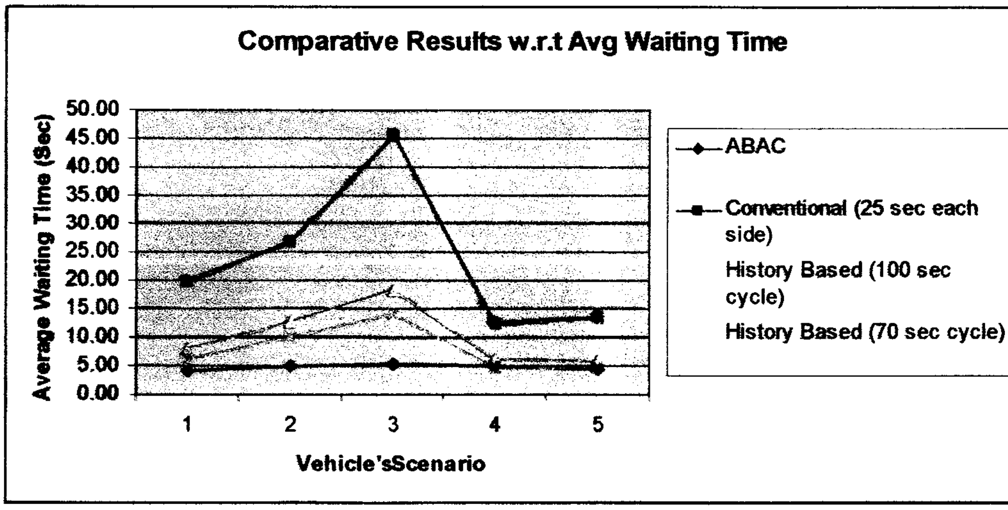

Figure 25. Graph for Independent Traffic Signal Scenario 3

Discussion

In above graph, again none of the approach shows minimum average waiting time than our proposed approach. The range of the average waiting time of our approach is 4.16 to 5.29, conventional approach with 12.33 to 45.48, history based 100 sec cycle time approach with 5.83 to 18.10 and history based 70 sec cycle time approach with 4.57 to 14.16. The history based approaches shows equal or with very less distance in two cases while in remaining cases ABAC architecture represent the less average waiting time.

6.4 Independent Traffic Signal with Rescue Vehicle

Here, in this case a rescue vehicle appears from any side of the independent traffic signal. Priority of the signal from where rescue vehicle appears will have the highest priority even the number of vehicles on that side is greater than other signals.

Scenario

Vehicles at signal (1, 2, 3 and 4) = 9, 7, 8, 6

Opening signal sequence (Before appearance of rescue vehicle) is 4, 2, 3 and 1

However, a rescue vehicle appears from signal 4 when signal 2 opens and after its appearance the priority of the side 4 will be increased. Due to this reason, signal 4 opens before signal 3 even has larger number of vehicles.

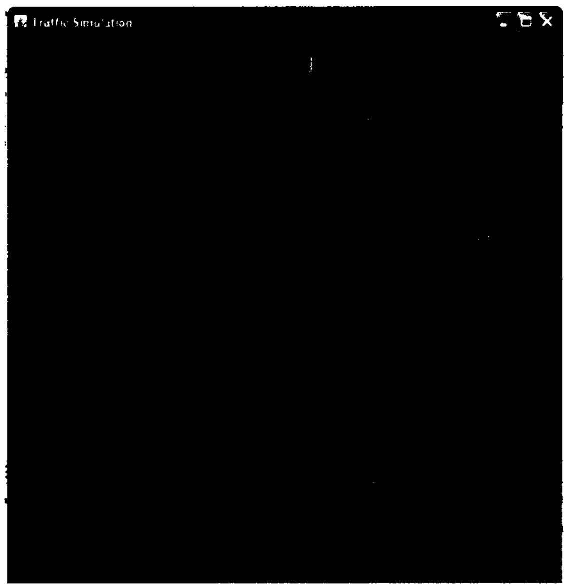

Figure 26. Independent Traffic Signal with Rescue vehicle Scenario

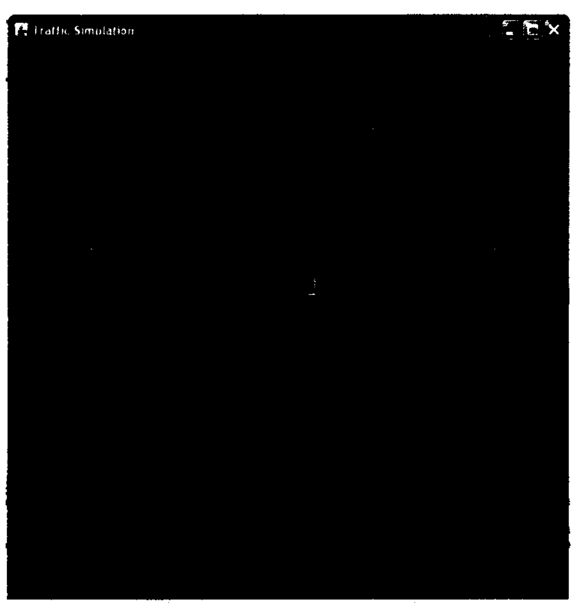

Figure 27. Rescue Vehicle in Independent Traffic Signal

Discussion

The opening signal sequence was 4, 2, 3 and 1. However, when signal 2 was opened the rescue vehicle appears from signal 4. After the appearance of rescue vehicle, priority of the side 4 will be increased. Due to this reason, signal 4 opens (even has larger number of vehicles than signal 3) before signal 3.

6.5 Mutual-Dependent Traffic Signals

In the above section we have discussed the simple and independent traffic signal scenarios. Here, we are discussing the more complex form of the traffic signals i.e. dependent signals. Two signals which are connected and close within the range of 50 meters are said to be dependent traffic signals. In case of dependent traffic signals, the traffic flow of one signal affects the other signal and vice versa as shown in Figure 28.

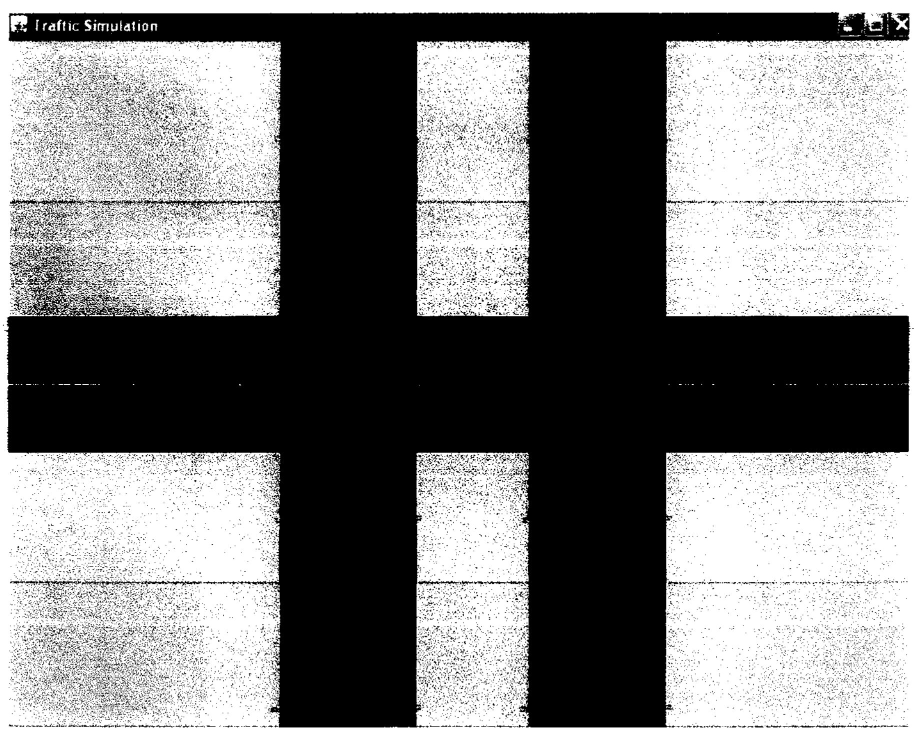

Figure 28. **Dependent Traffic Signal Scenario**

In this scenario, signals will be opened according to their number of vehicles waiting at some signal as well as their dependent signal's traffic. The independent traffic signal will be opened according to the algorithm as discussed in chapter 4 while the opening sequence of the dependent signal will be according to their number of vehicles (traffic flow waiting at current signal as well as dependent). Once again the dependent signal opening sequence will be from signal having less to larger number of vehicles.

6.6 Interconnected Traffic Signals with Rescue Vehicle

Here, we are considering the case where the rescue vehicle appears on interconnected signals as shown in Figure 29. In this case not only the priority of the signal from where rescue vehicle is increased but also the information about the rescue vehicle will be sent to the DM agent of next signal. After getting this information, the next signal will modify its signal opening sequence in a way so that rescue vehicle passes away without any delay.

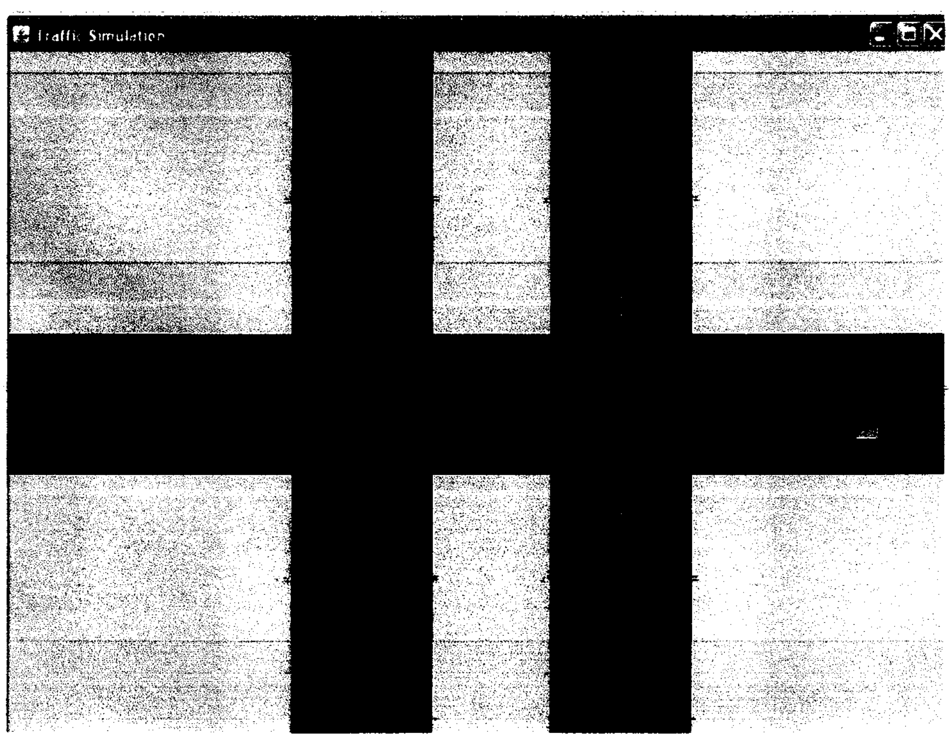

Figure 29. Dependent Traffic Signal with Rescue vehicle Scenario

In this scenario, rescue vehicle appears from the signal 6 which has larger number of vehicles than the other three sides. However due to the emergency vehicle, priority of this signal increases and signal will be opened with 10 sec more than the actual allocated time and DM agent of this signal will send information about rescue vehicle to the next signal. After getting the information, DM agent of that signal arranges the smooth pass of the rescue vehicle on first priority and after its passage the signal will retain to the previous sequence.

Conclusion

The research proposes an autonomous agent-oriented traffic control system to mange and defuses congestions on roads. The suggested solution uses a hybrid approach where the traffic from each side is controlled through a single controller while the internal details are taken care of by autonomous agents. The proposed architecture controls the traffic with minimum human involvement. Furthermore, it is featured by the special treatment for the rescue and emergency vehicles. The architecture is to a large extent adaptive to the road conditions especially to the intensity of traffic stream. We have conducted the research to control the traffic signals intelligently and validated it through a simulation that is based on different traffic signal's scenarios and how it is handled in dynamic and intelligent way through our proposed architecture. The simulation is developed in JavaBeans IDE and to understand the results we have presented these in the form of graphs. The simulation results show that our proposed approach saves too much time and increase traffic flow. If we summarize the benefits, it would be less human effort and error chances, dynamic decisions as per traffic flow, robust system to handle traffic, easy driving and reduction in accidents that ultimately saves precious lives, maximum traffic flow even on small roads, reduction in pollution and noise pollution, less fuel consumption, better social environment and cost reduction.

Future Work

Additionally, the suggested ABAC traffic architecture can be integrated with other systems such as traffic bulletin boards to share and update traffic congestion information with the public. These traffic boards can be placed before main roads to show the traffic congestion that will be helpful for drivers to change their route if road is blocked or experiences already heavy traffic.

References

Alagar, V. S., Muthiayen, D. (2003). *A Rigorous Approach to Modeling Autonomous Traffic Control Systems.* The 6th International Symposium on Autonomous Decentralized Systems (ISADS), Italy, pp 193-200.

Albagul, A,. Hrairi, M., Wahyudi. Hidayathullah M. (2006). *Design and Development of Sensor Based Traffic Light System*, American Journal of Applied Sciences 3 (3): 1745-1749.

Bernon, C., Capera, D., Mano, J-P. (2008) *Engineering Self-Modeling Systems: Application to Biology,* Int. Workshop on Engineering Societies in Agents World, Springer-Verlag, LNCS.

Casey, M. (2002). *MPEG-7 Sound Recognition Tools*, Mitsubishi Electric Research Labs, Cambridge, Unites States of America.

Champion, A., René Mandiau, Christophe Kolski, Alexandre Heidet, Andras Kemeny. (1999). *Traffic Generation with the SCANeR II Simulator: Towards Mulit-Agent Architecture.* Proceedings of the first Driving Simulation Conference, pp 311-324.

Cheung, S., Kamath, C. (2005). *Robust Background Subtraction with Foreground Validation for Urban Traffic Video,* EURASIP Journal on Applied Signal Processing, Volume 14, pp 1-11.

Cheung, S., Kamath, C. (2004). *"Robust techniques for background subtraction in urban traffic video,"* Video Communications and Image Processing, SPIE Electronic Imaging, San Jose.

Chris, A.. (2005). *Organizational Principles for Multi-Agent Architectures.* Series: WSSAT – Whitestein Series in Software Agent Technologies. Berlin; ISBN 3-7643-7213-2.

Horn, P. (2001). *Autonomic Computing: IBM's Perspective on the State of Information Technology*, IBM Journal Paper.

Huang, Y,. (2006) *Design of Traffic Light Control Systems Using Statecharts*, The Computer Journal, Vol. 49 (6).

Kasun, N., Hewage, J., Ruwanpura Y. (2004). *Optimization of Traffic Signal Light Timing Using Simulation.* Winter Simulation Conference, pp. 1428-1433.

Kephart, J. O., Chess, D. M. (2003). *The Vision of Autonomic Computing, Computer* 36(1), pp 41-50.

Lee, H., Lee, T. (2008). *A fast algorithm for measuring traffic vehicle parameters.* Proceedings of the 7th International Conference on Machine Learning & Cybernetics, Kunming, pp 3061-66.

Liu, X., Fang, Z. (2007). *An Agent-Based Intelligent Transport System.* 11th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Pp. 304-315.

Luck, M., McBurney, P., Preist, C. (2003). *Agent Technology: Enabling Next Generation Computing.*

Markus, C. H., Julie A. M. (2008). A *survey of Autonomic Computing Degrees, Models, & Applications.* ACM Computing Surveys, Vol. 40, No. 3, 2008.

Mueller, E. A. (1970). *Aspects of the history of traffic signals,* IEEE Trans on Vehicular Technology, 19 (1), 1970, pp 6–17.

Pour, G. (2006). *Expanding the Possibilities for Enterprise Computing: Multi-Agent Autonomic Computing.* 10th IEEE International EDOCW, pp. 33 – 33.

Ranjini, K., Kanthimathi, A., Yasmine, Y. (2011). *Design of Adaptive Road Traffic Control System through Unified Modeling Language,* International Journal of Computer Applications (0975 – 8887), Volume 14– No.7.

Sadek, A., N. Basha. (2006). *Self-learning intelligent agents for dynamic traffic routing on transportation networks,* in Proceedings of the 6th International Conference on Complex Systems (ICCS), Boston, MA.

Sam, R., Olena, T., Mennell, W. (2001). *An Agent Architecture for Vehicle Routing Problems.* SAC ACM 200l.

Shamshirband, S,. Shirgahi, H,. Gholami, M., Kia, B. (2008) *Coordination between Traffic Signals Based on Cooperative*, World Applied Sciences Journal Volume 5 (5): 525-530.

Stuart, R., Peter, N. (2008). *Artificial Intelligence, A Modern Approach,* 2nd Edition, ISBN No. 81-7758-367-0.

Tavladakis, K., Voulgaris, N. C. (1999). *Development of an autonomous adaptive traffic control system.* The European Symposium on Intelligent Techniques, Greece.

Tian, J., Tianfield, H. (2003). A *Multi-agent Approach to the Design of an E-medicine System*, Springer-Verlag Berlin Heidelberg.

Van, C. (2005). *Organizational Principles for Multi-Agent Architectures.* Series: WSSAT – Whitestein Series in Software Agent Technologies, ISBN 3-7643-7213-2.