

Design of Adaptive Road Traffic Control System through Unified Modeling Language

K.Ranjini
Department of CSE
Einstein College of Engg
TamilNadu,India

A.Kanthimathi
Department of CSE
Einstein College of Engg
TamilNadu,India

Y.Yasmine
Department of CSE
Einstein College of Engg
TamilNadu,India

ABSTRACT

In recent years, Unified Modeling Language (UML) has become most popular among modeling languages. UML is commonly used in the design and implementation of any system and software architectures. UML models help to achieve functional and non-functional requirements of the system. Furthermore, UML tools have enabled the creation of source code from UML diagrams in order to initiate the programming phase of building software. However, due to lack of clearly defined semantics in UML, creating source code from UML models have become challenging. Hence different UML diagrams have been used to represent the functionality of the system. The main objective of this paper is to model an Adaptive Road Traffic Control System using UML.

Traffic congestion is an ever increasing problem in towns and cities all over the world. Local authorities must continually work to maximize the efficiency of their road networks and to minimize any disruptions caused by accidents and events.

In this paper we proposed a UML model for an Adaptive Road Traffic Control system which provides a technique for controlling the traffic in highway network using signals that are automatically controlled by detectors. It coordinates the operation of the traffic signals in entire area (city or town) to give good progression to vehicles through the road network. Whilst coordinating all the signals, the signal timing fluctuates throughout the day, responding intelligently and continuously as per the changes in traffic flow. It removes the dependence of less sophisticated systems on signal plans, which have to be expensively updated.

Keywords

UML model, Signal plans, Adaptive Road Traffic Control, Detectors

1. INTRODUCTION

Nowadays Object Oriented Software Development process is widely used in the Software Industry. The emergence of Object-Oriented Programming has heavily contributed toward a standardized method of modeling known as the Unified Modeling Language (UML). In recent years, UML has become synonym for software modeling and is commonly used to model the software architecture as per the requirements. UML includes a set of graphic notation techniques to create visual models of software-intensive

systems. Source code can be easily be generated with the help of different UML diagrams for building the software. The correctness of source code depends on the UML specification which needs to be standard, complete, precise, and unambiguous. A good UML specification leads to clearly defined semantics and an efficient code can be generated. The present work ARTC is based upon the Road Traffic Control (RTC) System which is explained below.

2. EXISTING ROAD TRAFFIC CONTROL SYSTEM

In urban areas where traffic signals are nearby, the co-ordination of adjacent signals is important and gives great benefits to road users by increasing the utilization per unit time in the peak hours. Coordinating signals over a network of conflicting routes is much more difficult than coordinating along a single route. [1]

Early work developed off-line software to calculate optimum signal settings for a signal network. It can be used to compile a series of fixed time signal plans for different times of day or for special recurring traffic conditions.

Traffic control systems like traffic signal controllers, traffic blinkers, solar traffic blinker, etc. are used as an independent system at isolated intersections or as part of a synchronized chain of controllers for coordinated control of traffic.[2]

Preparing such signal plans requires traffic data to be collected and analyzed for each situation and time of day. This is time consuming, expensive and also requires repetitive surveys to understand the dynamics in traffic flow. Unless plans are updated frequently as traffic increases and patterns change rapidly the signals become less efficient.

3. PROPOSED APPROACH

The existing systems are quite complex and become obsolete in short time durations. In order to adapt the changing demands of speed and efficiency, a reliable software system which may handle the situation dynamically is needed. For this purpose a model of Adaptive Road Traffic Control (ARTC) system is designed in UML which can be converted into a Real Time system.

Software architecture based on UML models will help in handling complexities and drawbacks of existing RTC

systems and also help to better understand the domain. UML is the de-facto standard visual modeling language which is a general purpose, broadly-applicable tool supported, industry standardized modeling language which offers an extensive set of diagrams for modeling. The complexity of the problem domain requires extensive efforts for the clarification of the initial problem statement. Moreover, due to the extremely long lifespan of RTC systems, stable and robust analysis models enabling the integration of new operational scenarios are needed which can be efficiently obtained using UML models.

3.1 Adaptive Road Traffic Control System

To overcome the problems of RTC, the concept of a demand responsive adaptive traffic control system is introduced. In the design of an ARTC system UML will help to meet safety, reliability, availability, and robustness as demanded in an environment of steadily increasing traffic and it increases the availability of the resource and decreases the delay. The code obtained using the UML models are optimized which is one of the main requirements for the design of a RTC system.

An online computer will continuously monitor the traffic flows over the whole network using detectors in various places, fed the flows into an on-line model, and used the output from the model as input to its signal timing optimizers. These optimizers made a series of frequent small adjustments to signal timings to minimize the modeled vehicle delays throughout the network.

It also provides an important tool in the traffic manager's toolbox for managing the highway network. It coordinates the operation of all the traffic signals in an area to give good progression to vehicles through the network. Whilst coordinating all the signals, it responds intelligently and continuously as traffic flow changes and fluctuates throughout the day. It removes the dependence of less sophisticated systems on signal plans, which have to be expensively updated.[3]

3.2 Working principle

The Traffic controller which is an online computer will continuously monitor the traffic. The traffic controller must be linked to the detectors. The detector store house will store the information about the detector locations, physical layout of the road network and how the traffic signals control the individual traffic streams.

Any adaptive traffic control system relies upon good detection of the current conditions in real-time to allow a quick and effective response to any changes in the current traffic situation. Detectors are normally required on every

4.1 UML Usecase Diagram:

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

link. The location is important. To provide good information in advance of the vehicles' arrival at the stopline the detectors are usually positioned at the upstream end of the approach link.

Information from the detectors is input to the traffic controller, which models the progression of the traffic from the detector through the stop-line. It takes due account of the state of the signals and any consequent queues.

When vehicles pass the detector, the system will receive the information and converts the data into its internal units and uses them to construct "Cyclic flow profiles" for each link. Vehicles will reach the link at a normal speed and join back the queue if Red signal is on otherwise discharge from the stop-line.

The data from the model is then used by three optimizers which are continuously adapting three key traffic control parameters – the amount of green for each approach at each junction, the time between adjacent signals and the time allowed for all approaches to a signaled intersection. These three optimizers are used to continuously adapt these parameters for all intersections in the controlled area, minimizing wasted green time at intersections and reducing stops and delays by synchronizing adjacent sets of signals. The operation of the optimizers provides the necessary combination of responsiveness to traffic fluctuations and the stability to maintain coordination. The split optimizer optimizes every stage change, the offset is optimized each signal cycle for every node and the cycle time for each region is optimized once in every five minutes or once in every two and a half minutes when required to respond to rapid flow changes.[1]

ARTC signal timings evolve as the traffic situation changes without any of the harmful disruption caused by changing fixed time plans on more traditional urban traffic control systems. By the combination of relatively small changes to traffic signal timings, the system responds to short term local peaks in traffic demand, as well as following trends over time and maintaining constant co-ordination of the signal network.

4. ARTC ARCHITECTURE

System architecture is a set of design decisions. These decisions are technical and commercial in nature. To meet the functional and nonfunctional requirements of the above said ARTC system it is necessary to model the complete ARTC system by the use of UML. Different types of diagrams are designed and described below in brief:

The usecase diagram for the ARTC system is shown in the figure (Figure 1).

The usecase diagram has two actors namely detector that is used to detect the vehicles and the database that is used to store the traffic flow information. It describes only three usecases.

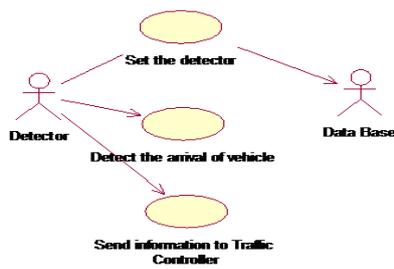


Fig 1 : Use Case Diagram for ARCTC

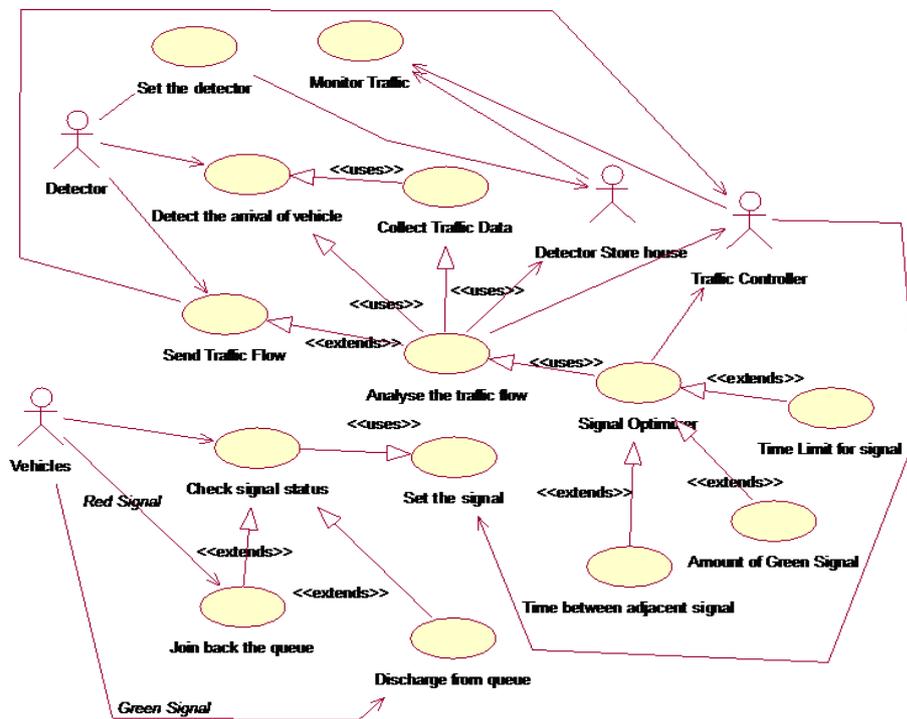


Fig 2: Extended Usecase Diagram

This usecase diagram can be used in the initial development. But it can be extended to include all the needed details. The figure (Figure 2) shows the extended usecase and it describes the complete detail of an Adaptive Traffic Control System which includes four actors. They are the detector that is used to detect the vehicles in the road, Traffic Controller that continuously monitor the traffic flow, Detector Store House that stores the details about the detector and the vehicle that passes over the detector and/or waits for the signal.

The usecases are self explanatory and they represent the main function in an ARTC system. The highlight of the system is the three optimizers which reduces the wastage of green signal and adaptively change the cycle of the signal based on the traffic in each direction.

4.2 UML Class diagram

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Since the class diagram is the main building block in object oriented modeling, It is used both for general conceptual modeling of the application and for detailed modeling; translating the models into programming code. [16]

The class diagram (Figure 3) describes the classes involved in ARTC system. Traffic controller is the main class which coordinates the other classes. The detector class

describes the physical location of the detectors. It also shows the count of vehicles passed over during a particular time.

The next class is optimizer, which is the parent of other three optimizer classes namely Signal Optimizer,

Time Limit Optimizer and Adjacent Signal Optimizer. The Road class contains the information about the roads and their average traffic flow rate.

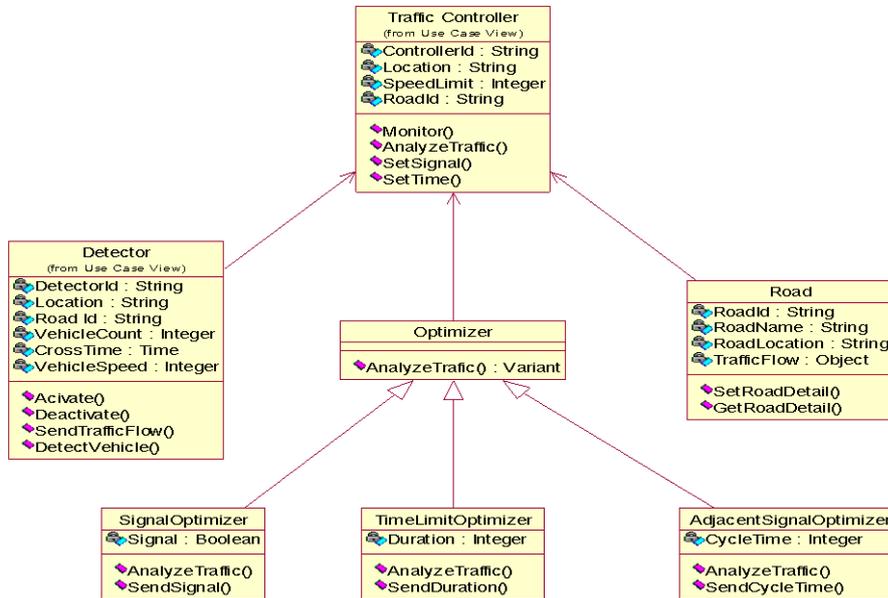


Fig 3 : UML Class Diagram



Fig 4: State Chart diagram for Vehicle Object

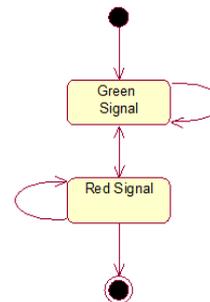


Fig 5: State Chart diagram for the signal object

4.4 Activity Diagram

Activity diagram depicts high-level business processes, including data flow, or to model the complex logic within a system. [4]

The figure (Fig 6) represents the various activities that take place within the ARTC system. According to this model, the detector will detect the vehicles that pass over by and it sends the information to the traffic controller. Then the traffic controller will consult the three optimizers to produce an

optimized result. Depending on the information received, it will set the signal and the time limit for the signal. The vehicles will check the signal and discharge if it is green otherwise it will join back the queue and wait for the signal.

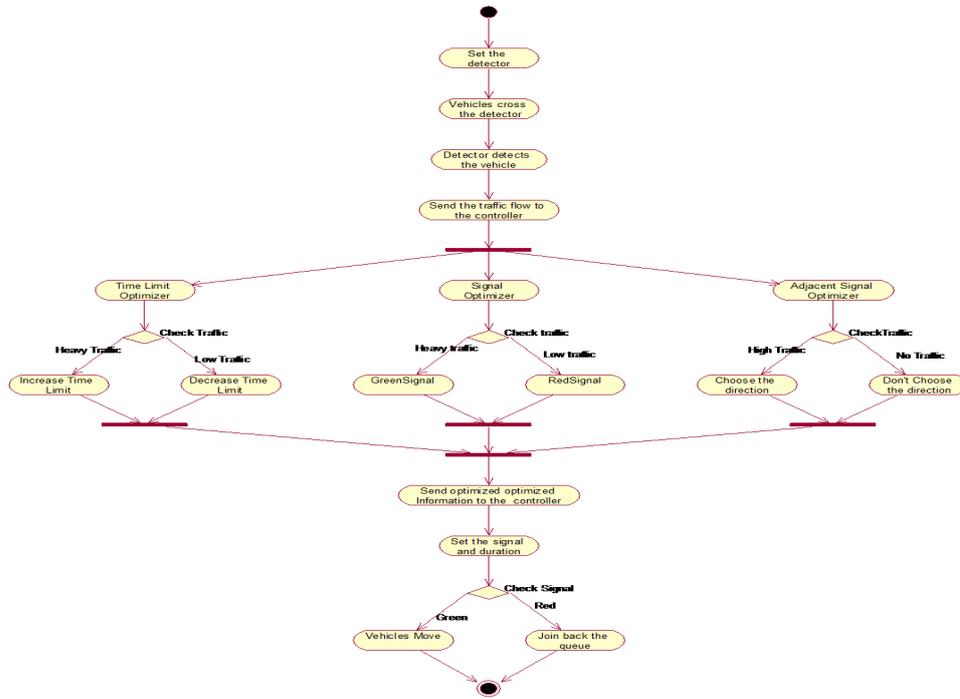


Fig 6: Activity Diagram for ARTC System

4.5 Interaction Diagram

Interaction diagrams model the behavior of usecases by describing the way groups of objects interact to complete the task. The two kinds of interaction diagrams are sequence and collaboration diagrams.

4.5.1 Sequence diagram:

Sequence diagrams demonstrate the behavior of objects in a usecase by describing the objects and the messages they pass. The diagrams are read left to right and descending. The figure (Fig 8) shows that an object of class vehicle starts the behavior by sending a message to an object of class detector. Then the detector object send message to the traffic controller object. This object in turn send message to the optimizer. The optimizer object will respond by sending return message. Finally the Traffic Controller object will send the command to the Signal object

4.5.2 Collaboration diagram:

Collaboration diagram shows the relationship between objects and the order of messages passed between them. The objects are listed as icons and arrows indicate the messages being passed between them. The numbers next to the messages are called sequence numbers. As the name suggests, they show the sequence of the messages as they are passed between the objects. There are many acceptable sequence numbering schemes in UML. (Fig 8)

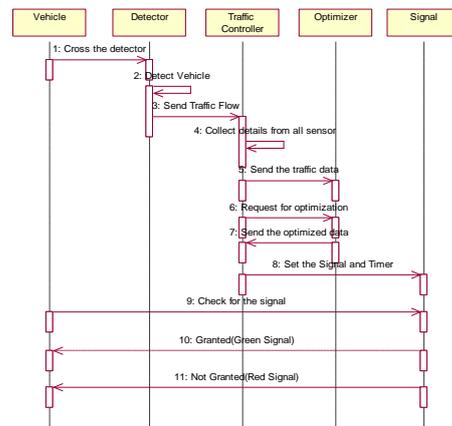


Fig 7: Sequence Diagram for ARTC System

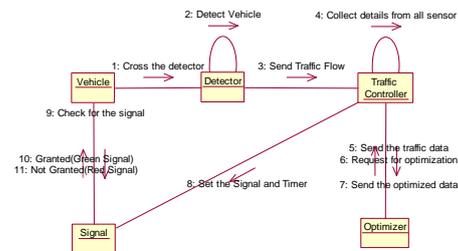


Fig 8: Collaboration Diagram for ARTC

5. CONCLUSION

In this paper, we have investigated the problems associated with the traditional Road Traffic Control and we proposed a solution to this problem by introducing Adaptive Road Traffic Control system which may react dynamically based on the situation. We also designed a complete UML modeling for the ARTC which is very efficient, effective and useful. This ARTC model can be utilized by other interested researchers to develop further in real time; can be implemented by the government to improve the traffic system for reduced congestion in RTC system.

6. REFERENCES

- [1] Advice Leaflet 1: The “SCOOT” Urban Traffic Control System : <http://www.scoot-utc.com>
- [2] Traffic Control System: <http://www.trafficsignals.co.in/traffic-control-systems.htm>
- [3] Shafeeq Ahmad, Vipin Saxena, “Design of Formal Air Traffic Control System Through Uml”, Ubiquitous Computing and Communication Journal: www.ubicc.org/files/pdf/formalATC_287.pdf
- [4] Martin Fowler, ”UML Distilled- A Brief guide to the standard object modeling language”, Second Edition, Pearson Education.
- [5] Intelligent transportation system: http://en.wikipedia.org/wiki/Intelligent_transportation_system
- [6] D. Kinny, and M. Georgeff. Modelling and Design of Multi-Agent Systems. Intelligent Agents III, Springer, 1996.
- [7] Dr Vipin Saxena, Deepa Raj, “Local Area Network Performance Using UML”, International Journal of Advanced Networking and Applications Volume: 02, Issue: 02, Pages:614-620 (2010).
- [8] M. Wooldridge, N. R. Jennings and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. International Journal of Autonomous Agents and Multi-Agent Systems, 3,2000.
- [9] D. E. Herlea, C. M. Jonker, J. Treur, and N. J. E. Wijnngaards. Specification of Behavioural Requirements within Compositional Multi-Agent System Design. Proceedings of Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, pages 8-27, Springer, 199
- [10] Model-driven Architecture: <http://www.omg.org/mda/>
- [11] Rudolph, E., Grabowski, J., & Graubmann, P. Tutorial on message sequence charts (MSC). In Proceedings of FORTE/PSTV'96 Conference, 1996.
- [12] Rumbaugh, J.: OMT: The development model. JOOP Journal of Object Oriented Programming, pages 8-16, 76, 1995
- [13] Rumbaugh, J.: OMT: The dynamic model. JOOP Journal of Object Oriented Programming, pages 6-12, 1995. Agentis. <http://www.agentissoftware.com>.
- [14] Bauer, B. and Müller, J.P.: Methodologies and Modeling Languages, in: Agent-Based Software Development, Luck M., Ashri R. D’Inverno M. (eds.) Artech House Publishers, Boston, London, 2004.
- [15] Kleppe M., Warmer J., Bast W. MDA Explained – The Model Driven Architecture: Practice and Promise, Addison Wesley, 2003
- [16] Upcoming UML 2.0 Standard: <http://www.omg.org/uml>
- [17] Dr. Manuj Darbari*, Niranjan Srivastava, Siddhartha Lavania, Sunita B ansal. Information Modeling Of Urban Traffic System Using Fuzzy Stochastic Approach, Proceedings of the World Congress on Engineering 2010 Vol I, WCE 2010, June 30 - July 2, 2010, London, U.K.
- [18] Alessandra C, (2004), "A framework to simulate UML Models, Proceedings of the 2004 ACM Symposium on Applied Computing