

# Formal Modeling and Verification of Smart Traffic Environment with Design Aided by UML

Umber Noureen Abbas

Department of Computer Science  
COMSATS Institute of Information  
Technology Sahiwal  
Sahiwal, Pakistan

Nazir Ahmad Zafar

Department of Computer Science  
COMSATS Institute of Information  
Technology Sahiwal  
Sahiwal, Pakistan

Farhan Ullah

Department of Computer Science  
COMSATS Institute of Information  
Technology Sahiwal  
Sahiwal, Pakistan

**Abstract**—Issue challan in response to rules violation, LED (Light Emitting Diode) and Bridge components of this proposed Smart Traffic Monitoring and Guidance System are presented in this paper to monitor violation of rules, update users about traffic congestion through LED and to provide central hub to communicate with sensors to update server about the traffic situation. It involves the Wireless Sensors and actors to communicate with the system. The proposed components require fewer resources in terms of sensors and actors. Further sensors identify violation of rules through issue challan. Secondly, LED component provides information to users about the traffic situations. Thirdly, Bridge component is used to provide central hub to communicate with different components in the proposed model and to update the server. The proposed components of this model are implemented by developing formal specification using VDM-SL. VDM-SL is a formal specification language used for analysis of complex systems. The developed specification is validated, verified and analyzed using VDM-SL Toolbox.

**Keywords**—Formal specification; Formal analysis; VDM; Sequence Diagram; Issue Challan; LED

## I. INTRODUCTION

Traffic congestion is a big problem in developed countries. In advanced countries better transportation management system has the ability to handle complex situations. It is quite important to suppress traffic problems because it can contribute a lot in the development of a country. Traffic congestion can cause, accidents, wasting of time, minimize trade opportunity, increases energy consumption and inversely affect the education system and the impact of all these can be dangerous for living societies [2].

The main objectives of traffic management system should be the ability to reduce congestion, overcome traffic pollution and smooth traffic flow. Researchers have proposed different models to make traffic system robust, smarter and reliable. In order to maintain the change in demand of speed and efficiency, a reliable traffic system is required that can manage and monitor traffic in a dynamic environment. With the growth of population traffic becomes a critical issue, irregular traffic is a big problem in populated cities of Pakistan. To sort out these issues we have to develop a smart traffic management system in which data can be collected from sensors in real time, processed at the same time and update the system to remove

congestion. This paper has proposed a model and formalized few components in VDM-SL, which are discussed in [1]. Researchers have proposed different models with the involvement of different tools and technologies. These models have the capability of collecting road traffic information through wireless sensors and other employed devices. They collect information about vehicles, pedestrians and other moving objects in real time that can affect road network. This collected information can also be presented to pedestrians and vehicles so they can decide their route. We need all these things to minimize any distraction on roads that can cause to accident or disaster. Real-time systems can recognize the road traffic make certain decisions and predict the flow of traffic. This paper has also proposed certain functions in the model that works in real-time like to change the signal timings and to identify the occlusion. Developing intelligent traffic management system with the integration of IoT has more advantages than employing other related technologies. IoT save more cost, reliability, and safety than other mechanisms [21]. This proposed model and different other models employed IoT devices which have a number of RFIDs and wireless sensors that communicate with each other and sends data to the administrator. This is a challenge for future researchers to ensure the security of smart objects and their transmitted data so no intruders can attack on data [3]. The proposed traffic monitoring and guidance system provides an efficient way of monitoring traffic flow and it helps to avoid congestion, provide smooth traffic flow better utilization of resources and decrease the management cost.

Currently, this study is focused on the bridge, Issue Challan, and LED Information components in this system model. Bridge is the central location where all the wireless sensors from any location connect and communicate. Issue Challan is the procedure of issuing Challan to vehicles who violate the traffic rules. The last module is related to the procedure of changing information on the LED that is employed on roads to help drivers. The formalization of this system is done in Vienna development method specification language, some of the modules are already presented in a previous article and remaining is presented in this paper. This paper consists of different sections which are Introduction, Literature Review, Problem statement, Formalization of modules, Sequence diagrams of these modules and conclusion.

## II. LITERATURE REVIEW

The problem of traffic management is quite critical so different researchers have dealt with the problem and gave their solutions, indeed different intelligent traffic monitoring and controlling systems have been described and proposed as a result of their efforts. Pang et al. [19] proposed a system of traffic flow prediction based on a fuzzy neural network. Bhadra et al. [7] presented a technique based on applied agent-based

fuzzy logic for traffic, involving multiple approaches and vehicle movements. In [15] authors have proposed a technique that can integrate dynamic data into intelligent Transportation Systems. Patrik et al. [23] proposed a service-oriented architecture (SOA) for an effective integration of IoT in enterprise services.

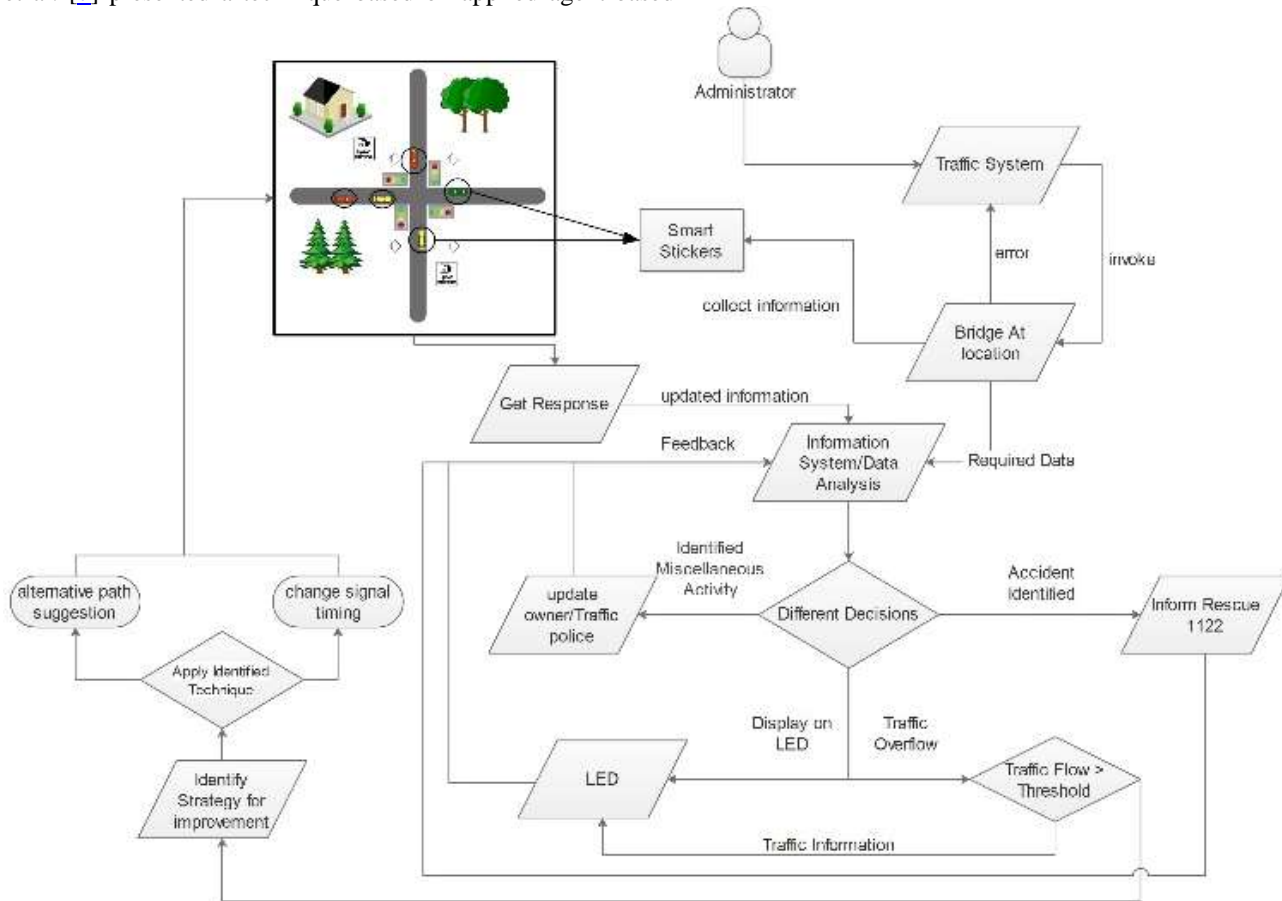


Fig. 1. Smart Traffic model with Design Aided by UML

Different researchers developed different paradigm for IoT composed of different intelligent systems. The domain of such intelligent systems in [4, 10, 17, 22] are health care, business inventories, smart environment, smart home, retail smart agriculture, supply chain logistics, monitoring electrical equipment. In this case of the intelligent transportation system is the earlier stage with respect to their needs [8, 20, 24, 25]. Various IoT smart systems such as FeDNet [16, 18], UbiComp [12] are using the simple message passing communication techniques. Such techniques using a large amount of energy and bandwidth.

Fortino et. al [11] have proposed cloud computing and architecture integrating agents techniques for developed reorganized smart objects within IoT. Godfrey et al. [9] have proposed the mobile based agent technique. This technique is not just handling the communication among the different devices, it also conduct searching for needed resources [9, 14, 18].

## III. MODEL OF TRAFFIC MONITORING AND GUIDANCE SYSTEM

In this proposed model Authors have explored main and important problems of traffic system. The major problems are traffic congestion, road accidents, identification of illegal and rules violating vehicles and emergency incidents. Another aspect is, large number of drivers are illiterate. They cannot read or effectively understand the traffic rules, signs and guidance provided on road. In this proposed model it provides guidance on road through Light Emitted diode (LED) at different points and intersections. In this study the focus is on display information on LEDs, issue challan on rules violation. LED display information related to current traffic state, future congestion prediction, path information, signal timings and other related information. Sensors installed at different point will scan the traffic situation on different intervals and then system will be updated with that information. Further, Bridge is another component to provide central hub for

communication among different components of the system. The Light Emitted Diodes will provide the display information to every user.

Issue challan will active in response to violation of rules, challan will be issued by the administrator and sent to owners mobile also system will inform nearby traffic wardens about violation. During processing system will also detect any miscellaneous activity and inform to related department if detected. In a previous study, Authors have described the traffic signals and smart stickers, traffic signals are controlled by the system automatically and it will also update the signal timing according to the situation.

#### IV. UML SEQUENCE DIAGRAMS

##### A. Traffic System Sequence Diagram

In a typical traffic system, all the equipment's are required to be online. The reason is system can request anytime. A mature information system has strong behavior. Therefore the work is done in a sequence. So that this diagram is called as sequence diagram. The first sequence of operation is related to maintenance of traffic system and at initial point any of the sequence can be initiated that belongs to system. First, the admin make a request for system. After this, the system makes a request for information system that collects all the information that is necessary from sensors. Now at this point the sensors receive the raw fact and send it to information system. Information system is required to process that coming information in real time and send back to LED or system admin. So the loop represents the working of information system.

This all due to sensors because information sending and receiving at the same time is much difficult task. In this way it identifies the flow of traffic. The processed information then used for changing the signal timings or other necessary actions.

The analyzer is checking or comparing the updated traffic information with the previous signal timings either there is any need to change the signals or not. After analysis the data it again sends to system and all this happens in fraction of second. The information is collected by sensors and stored in a system. If it is successful then make a normal feedback to the admin. Improvement centre also maintain history of situations and signals timings that perform well at certain time, this historical data is necessary while making a system automate. Now after completing all the stages admin have information that is completely correct.

To show the interaction between objects or sequential functionality of a system objects generally developers use sequence diagram. Sequence diagrams helps developers to develop that particular task easily, so developers think sequence diagrams are meant for them [5]. However traffic system sequence diagram can find the sequence between admin and system. First of all, admin give command to get information from sensors. System communicate and gather information from activate sensors because the system is using different types of sensors to get different information so system gather information from different type of sensors.

Unified modeling language have another technique to model the system, it is called class diagram. A class diagram represents the static behaviour of any system, in previous diagrams author was looking into dynamic behaviour of the system. Class diagram is a modeling technique in which anyone can represent the structure of system with its attributes and functionalities associated with it. All the previous functionalities are represented in this diagram and it also show the relationship between them [6] how a complete system is linked from inside and what information sharing is going on can be seen with this diagram. Class is also one way to represent a system as object oriented system.

##### B. LED Information Update Sequence Diagram

Sequence diagram is a timely or sequence based system interaction diagram that shows, how many process will be in the current system module and either they are communicating with each other or not. Which operation comes first and which comes later, how and why a process is dependent on other process [13]. It provides the sequence between processes, in sequence diagram interaction between operations arranged in time sequence. As discussed earlier in this scenario the pattern of reading a sequence diagram is standard way of reading from left to right and top to bottom. The figure shows the admin start the processing by sending the command to traffic system.

The bridge collects all information from the desired location and sends it to system for processing and then system put back that processed information on the LEDs. In this way the information is updated and displayed on LED. Here there are two cases which are:

Case I: If there is a request for update information then the admin give a request to system to call a bridge and ensure that all the information that are collected by the bridge is correct and make a request to display updated information on LEDs.

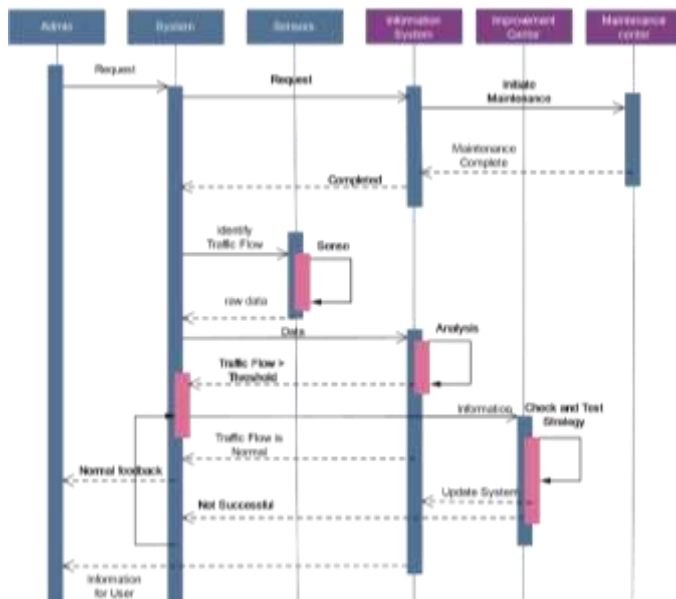


Fig. 2. Traffic System Sequence Diagram

Case II: If there is no request for update information then the old information is displayed on LEDs.

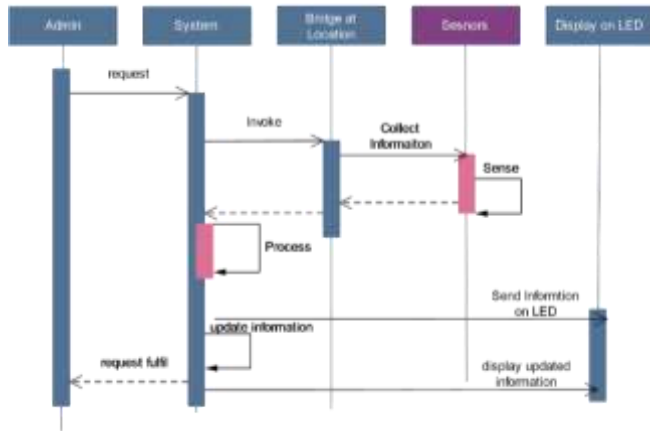


Fig. 3. LED Information Update Sequence Diagram

### C. Issue Challan Sequence Diagram

Interactive diagrams model define the behaviour of system in a way that when different group of users interacts with the same system, it can give different output behaviour according to users group. There are two diagrams in common for modeling interactivity of system those are collaborative and sequence. UML sequence diagram shown below, different lines represent different functions applied to system. Sequence diagram read from left to right (fig) shows that the object of system activate the behaviour by sending the request at bridge location. Then the object bridge at central location collect the information from sensors, then these sensor can detect the vehicle its speed, geo-location, direction and other metadata required to make decisions. The collected data will be processed by information system, if the information relates with miscellaneous activity or rule violation then, the system updates its database and send the situation and all other necessary information to nearest traffic police representative. Related information may also be sent to vehicle owner.

The word violation means to break the law for some reason or no reason. The violation might be high speed, don't fasten the seat belt and wrong overtaking etc. How to reduce the traffic violations? If the violation occurs then where it is occurring? Identify it and how to engage it? These are basic questions to be answered in a well-suited manner first of all the system makes a request for the nearby bridge at a central location. Sensor device has short range as compared to bridge. If we say the range is enough for sensor then the problem will, all the sensors cannot make direct connection with the server. Bridge is basically collector that is why the system use bridge is to group sensors that receive all the information from sensors and directly connect to the server. Working of bridge is like a Wi-Fi device to receive and collect the signal from host devices or server then send it to all main and major devices that are attached or connected with Wi-Fi. So, in the same way, the bridge collects all the information from the sensors and then convey this to server. The bridge collects the information in the form of raw data. Then the data is forwarded to information system. At this stage, the information system identifies either the activity is miscellaneous or not. Miscellaneous activities

involve all types of rules violation if it identifies then that activity will be reported to police representative and system admin.

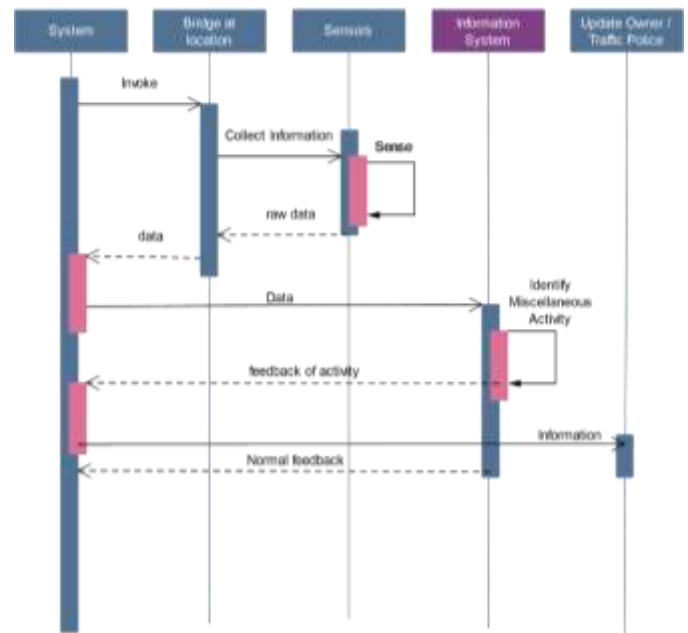


Fig. 4. Traffic Rules Violation Sequence Diagram

## V. FORMALIZATION OF MODULES

### A. Bridge

Bridge is a device that is used to connect all the sensors with the server. All the sensors cannot connect directly with the server, because it will make a mess. So the system needs a central point (bridge) where all the sensors of a certain location connected and then bridge will be connected to server.

In the formal specification of bridge, the first part this system has is types in which all the variables are declared. The system used different variables like bridge status that is quote type, Bridge, sensor information that is the token type, location, and sensor that is a string. Other variable is traffic situation and the last one is restricted area. The system used these variables for storage of different values as name suggests traffic situation will store the current state either it is jammed congestion, smooth or accident. Restricted area store different areas that are restricted. Same as location will store the locations.

```

types
String = seq of char;
bridgestatus =
<AVAILABLE>|<BUSY>|<IDLE>|<ERROR>;
Bridge::name : String
      Status : bridgestatus;
Sensorinformation = token;
Location = String;
Sensors = String;
trafficSituations =
<JAMMED>|<CONGESTION>|<SMOOTH>|<ACCIDENT
>;
restrictedAreas = set of token;
    
```



The system is not assigned with any predefined value so the values portion is empty. The most important part of any specification is its state. State defined the starting behaviour of any program and it gave values to different variables that remained in the memory. Different variables defined in the state with their data types, the state also has invariant and initialization, invariant is a condition that must be satisfied before the declaration of object, and initialization define the values of all the variables in the state on declaration. So the system is checking Bridge and restricted area must not be empty.

```
state theBridge of
loc : map Sensors to Location
bridge : set of Bridge
getinformation : map Sensors to Sensorinformation
restrictedArea : set of restrictedAreas
currentTraffic : set of trafficSituations
inv mk_theBridge(-,b,-,r,-)= (b <> {}[2]) and (r <> {})
init i==i=mk_theBridge({|->},{},{|->},{},{|})
end
```

The system can perform different operations on Bridge; these are add sensor, get information and status update. First Operation is ADD SENSOR that has two inputs, first is a sensor that is to be added in network and the other is location where the sensor is to be deployed. The precondition is checking, is this sensor is new or already used, so if the sensor is not already used map that sensor to the location.

```
operations
Addsensor(sensorin:Sensors,locationIn:Location)
ext wr loc:map Sensors to Location
pre sensorin not in set dom loc
post loc = loc munion{sensorin|->locationIn};
```

Another important operation is GET INFORMATION FROM SENSORS. It has also two inputs and it will give the information from the sensors after sensing. The precondition is checking the information that is not already stored in the system and the sensor is in the domain of location. It will get the information and store in the get information variable.

```
Getinformtionformsensors(getinformationin:Sensorinformatio
n, sensorin:Sensors)
ext wr getinformation: map Sensors to Sensorinformation
wr loc:map Sensors to Location
pre (getinformationin not in set rng getinformation) and
(sensorin in set dom loc)
post getinformation= getinformation munion{getinformationin
|-> sensorin};
```

SENSE INFORMATION operation defining the context of information from where it is coming, giving the location, the bridge and the type of information is to be sense. The precondition is used to check either the bridge exist, post condition is used to check either the bridge status is available or the traffic situation is also known. If all these conditions are true, then take information from the sensors and store it to get information variable.

```
SenseInformation(nameIn:String,statusIn:bridgestatus,locatio
nIn:restrictedAreas,sensorIn:Sensorinformation,anyTrafficSitu
tion: trafficSituations)
ext wr restrictedArea : set of restrictedAreas
wr getinformation: map Sensors to Sensorinformation
wr bridge : set of Bridge
```

```
wr currentTraffic: set of trafficSituations
pre (mk_Bridge(nameIn,statusIn).Status <> <ERROR>)
post (mk_Bridge(nameIn,statusIn).Status =<AVAILABLE>
and (locationIn in set restrictedArea)or(anyTrafficSituation in
set currentTraffic ) or getinformation = getinformation
munion{getinformation|->sensorIn});
```

The last operation is STATUS UPDATE that is used to update the status of bridge. It is necessary to maintain all the devices so during maintenance, bridge status might be changed to busy or error and update status might be converted back to active.

```
StatusUpdate(statusIn:bridgestatus,bridgeIn:Bridge)
ext wr bridge : set of Bridge
pre true
post bridgeIn.Status=statusIn;
```

## B. Issue Challan

Issue challan is common when somebody breaks the rule. This specification defined different situations when somebody got challan with different processes involved in it. The system has used different variables, vehicle owner that is a composite type, reasons that stored the reason of a challan. Sensors information that is either information about the vehicle or information about the reasons for challan. The system also store location.

```
types
String = seq of char;
Vehicleowner ::id:String
License:String;
Resons=<OVERSPEED>|<CROSSREDSIGNAL>|<TAXESO
VERDUE>;
Challan=String;
Sensorinformation=<VEHICLEOWER>|<RESONS>;
Location = String;
Sensors = String;
```

In the values the system has used speed limit, its type is integer and value is sixty, it may be used inside the city for checking over speed.

```
values
SPEEDLIMIT:int=60
```

The state is an important part of any specification, in the invariant of this state, the system is checking for reasons, because if the system is issuing a challan then there must be a reason for it. So the cardinality of reasons must be equal to the cardinality of challan. The detail of challan will come later and for the initialization, yet all the variables are empty.

```
state issuechallan of
loc: map Sensors to Location
sensors : set of Sensors
getinformation: map Sensors to Sensorinformation
cardriver :set of Vehicleowner
resons: set of Resons
challan:set of Challan
inv mk_issuechallan(-,-,-,r,c)= (card(c)=card(r))and
(r={<OVERSPEED>} or r={<CROSSREDSIGNAL>} or r =
{<TAXESOVERDUE>})and( SPEEDLIMIT > 60 )
init i==i=mk_issuechallan({|->},{},{|->},{},{|})
end
```

Next part is operations that define the behaviour for any system. In this module the first operation is add sensor same as

in other modules management can add sensor to any location. The ADD SENSORS is used to add sensor and where to deploy it. For adding a sensor, management uses mapping function that Maps a Sensor to the given location.

```
operations
AddSensors(sensorin:Sensors,locationIn:Location)
ext wr loc: map Sensors to Location
wr sensors : set of Sensors
pre sensorin not in set dom loc
post loc= loc munion{sensorin/->locationIn};
```

The GET INFORMATION FROM SENSOR operation can be invoked manually or automatically. This method senses the information from the sensors and give it to the calling authority, for the input parameters.

```
Getinformationfromsensors(getinformationin:Sensorinformation, sensorin:Sensors)
ext wr getinformation: map Sensors to Sensorinformation
pre (getinformationin not in set rng getinformation) and (sensorin in set dom loc)
post getinformation= getinformation
munion{getinformationin /-> sensorin};
```

ISSUING CHALLAN is necessary to control the traffic on roads. To issue challan the system must have some reason, id and license of the person.

```
Issuchallan(resonIn:Resons,idIn:String,licencIn:String)
ext wr cardriver :set of Vehicleowner
wr resons: set of Resons
wr challan: set of Challan
pre true
post (resonIn=<OVERSPEED>and (SPEEDLIMIT>60)or resonIn=<CROSSREDSIGNAL>or resonIn=<TAXESOVERDUE>) and challan = challan union {mk_Vehicleowner(idIn,licencIn)};
```

The system must have an option in it to check whether the person has paid his or her previous challan or not. So IS PAID CHALLAN method takes challan and check whether it is paid or not paid.

```
Ispaidchallan(challanIn: Challan)query:bool
ext wr cardriver :set of Vehicleowner
wr challan: set of Challan
pre true
post query <=> challanIn in set challan ;
```

NUMBER OF ISSUE CHALLAN method used the total count of challans issued in a system to any user.

```
numberofissuechallan(number:int)
ext wr challan:set of Challan
pre true
post number = card challan ;
```

### C. LED Information

The Authors have deployed LEDs on roads for displaying information to users. The Authors have developed this module to set and edit the LED information. The first part of the specification is types that have variables declared in it then these variables are used in operations and state of LED. LED information variable store information that is currently displayed on the LED and if management wants to change the information they have to modify the information that is currently stored in it. Second variable the system use is

information about the traffic either the traffic is jammed, is there any accident and is there any construction going on or congestion. Next variable is related to information given to a user about the path.

```
types
String = seq of char;
ledInformation = set of String;
information
=<CONGESTION>|<JAMMED>|<CONSTRUCTION>|<ACCIDENT>;
Signalforvehicleowner
=<CONTINUE>|<SUGGESTIONPATH>;
SensorinformationonLED=<INFORMATION>|<SIGNAL>;
Location = String;
Sensors=String;
```

Next part of the specification states that defined the initialization of a class. The System has used invariant that triggers before the initialization and validates some conditions, and then initializes the variables according to values if given. In this situation, the system doesn't have any values given in the initialization.

```
state LED of
loc: map Sensors to Location
sensors : set of Sensors
getinformationofLED: map Sensors to SensorinformationonLED
ledInfo:set of String
newinfo : map String to String
inv mk_LED(-,-,l,n) == (card(l) > 0 and card(l) <= 30)and (l subset dom n)
init mk_LED(-,-,l,n)== l = {} and n={/->}
end
```

The first operation is ADD SENSORS, a method is defined to take a sensor and a location and it maps the sensor to the location.

```
operations
AddSensors(sensorin:Sensors ,locationIn: Location)
ext wr loc: map Sensors to Location
wr sensors : set of Sensors
pre sensorin not in set dom loc
post loc= loc munion{sensorin/-> locationIn };
```

GET INFORMATION FOR LED method takes input to display on LEDs from sensors. It gets information from sensor after system processed that information and then display on the LED, before the pre-conditions validates that this information is not already displayed on LED.

```
GetinformationforLED(getinformationin:SensorinformationonLED, sensorin:Sensors)
ext wr getinformationofLED: map Sensors to SensorinformationonLED
wr loc:map Sensors to Location
pre (getinformationin not in set rng getinformationofLED) and (sensorin in set dom loc)
post getinformationofLED= getinformationofLED
munion{getinformationin /-> sensorin};
```

ADD INFORMATION method takes information and displays it on the LEDs. This method is manual, when admin wants to display desired information on the LEDs then admin will use this method to send the string to LEDs.

```
AddInformation(newInfo:set of String)
ext wr ledInfo:set of String
pre newInfo <> {}
post ledInfo = newInfo union ledInfo;
```

REMOVE INFORMATION works same as the previous method. It is possible that LEDs are displaying some information one after another. Therefore, any information management wants to remove from the LEDs they have to pass it to this method.

```
RemoveInformation(removeInfo:set of String)
ext wr ledInfo:set of String
pre removeInfo <> {}
post ledInfo = removeInfo \ ledInfo;
```

As the name suggests CHANGE INFORMATION method change the text from LEDs, it replaces old information with the new information. Admin needs to change information occasionally so this method is very important.

```
changeinformation(oldinfo:String,lednewinfo:String)
ext wr newinfo : map String to String
pre lednewinfo not in set dom newinfo
post newinfo =newinfo ++{lednewinfo |-> oldinfo};
```

VI. FORMAL ANALYSIS OF MODULES

A. Formal Analysis of Bridge

Formal analysis of bridge is shown in Figure 5 and Table 1.

TABLE I. FORMAL ANALYSIS OF BRIDGE

Operation	Syntax Check	Type Check	Integrity Check
Addsensors	Y	Y	Y
Getinformationfromsensor	Y	Y	Y
Senseinformation	Y	Y	Y
StatusUpdate	Y	Y	Y

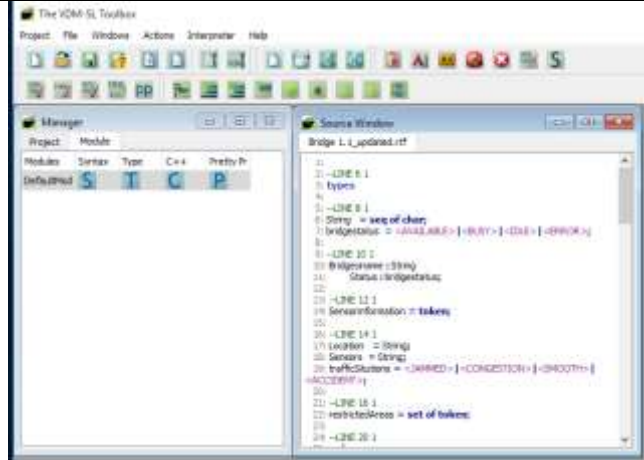


Fig. 5. Formal Analysis of Bridge

B. Formal Analysis of Issue Challan

TABLE II. FORMAL ANALYSIS OF ISSUE CHALLAN

Operation	Syntax Check	Type Check	Integrity Check
Addsensors	Y	Y	Y
Getinformationfromsensor	Y	Y	Y
Issuechallan	Y	Y	Y
IsPaidchallan	Y	Y	Y

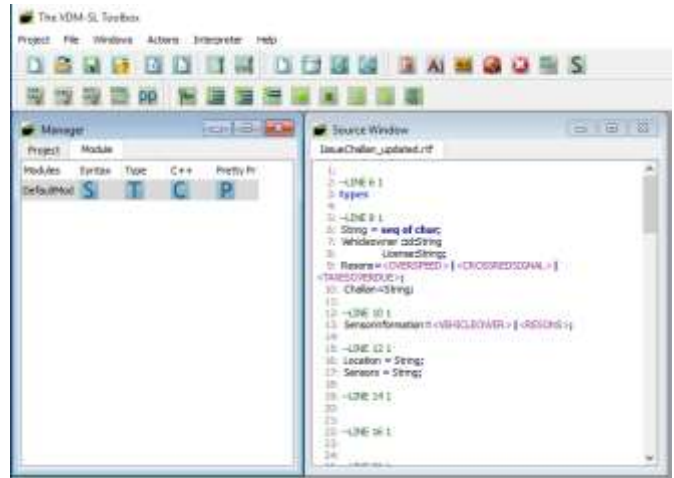


Fig. 6. Formal Analysis of Issue Challan

C. Formal Analysis of LED Information Update

The analysis is performed as shown in Table 3 and Figure 7.

TABLE III. FORMAL ANALYSIS OF LED INFORMATION UPDATE

Operation	Syntax Check	Type Check	Integrity Check
Addsensors	Y	Y	Y
Getinformationfromsensor	Y	Y	Y
Addinformation	Y	Y	Y
Changeinformation	Y	Y	Y
removeInformation	Y	Y	Y

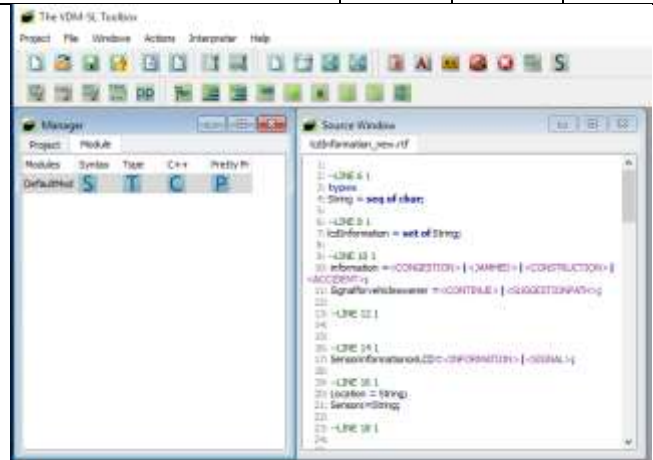


Fig. 7. Formal Analysis of LED Information Update

VII. CONCLUSION AND FUTURE DIRECTIONS

This paper presented modules of Smart Traffic Monitoring and Guidance System like issue challan, display information of traffic situation on LEDs, bridge and system maintenance. The goals of these modules are to provide information related to traffic, its rules and display information on LEDs at different locations on roads to guide users. Traffic rules violation monitored remotely through sensors, further these sensors collect information of vehicle owners and about those persons who violate the rules. First, LEDs are used to show rules and status of different roads, sensors are used to detect rules

violation, also sensors are used to detect traffic load and finally bridges are used to combine sensors from a certain location. The system needs to be maintained in its life cycle, so Authors have also proposed maintenance module in which system errors and bugs will be fixed and system values will be adjusted. Smart traffic system provides the information related to traffic situation to overcome the incidents. Formal methods based technique, i.e. VDM-SL is used to develop a formal representation of the proposed model. The proposed model components are validated, verified and analyzed using VDM-SL Toolbox. In a previous paper, the authors have already discussed the traffic signal, rescue services, and smart sticker services to provide the security and trouble-free environment.

In future, there is a need to design smart algorithms for this proposed Smart Traffic Management and Guidance System to enhance the functionality of sensors. This model can be connected to web domain in order to inform users on their desktop about traffic situations. Sensors can be connected with GSM (Global System for Mobile communications) that can be able to broadcast information through SMS (Short Message Service) to users which are in range. In future, this model can be enhanced to provide smart services in traffic management system in the country.

#### REFERENCES

- [1] Abbas, U.N., F. Ullah, and N.A. Zafar, Formal Model of Smart Traffic Monitoring and Guidance System. *International Journal of Computer Science and Information Security*, 2016. 14(6): p. 241.
- [2] Al-Heeti, K. and M.S. Al-Ani, Intelligent Traffic Light Control System Based Image Intensity Measurement. 2011.
- [3] Al-Sakran, H.O., Intelligent traffic information system based on integration of Internet of Things and Agent technology. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2015. 6(2): p. 37-43.
- [4] Asin, A. and D. Gascon, 50 sensor applications for a smarter world. *Libelium Comunicaciones Distribuidas*, Tech. Rep, 2012.
- [5] Bell, D., UML basics: The sequence diagram. IBM Corporation, Feb, 2004. 16: p. 15.
- [6] Bertini, R.L., M. Leal, and D.J. Lovell, Generating Performance Measures from Portland's Archived Advanced Traffic Management System Data. *Transportation Research Board*, 2002.
- [7] Bhadra, S., A. Kundu, and S.K. Guha. An agent based efficient traffic framework using fuzzy. in *2014 Fourth International Conference on Advanced Computing & Communication Technologies*. 2014. IEEE.
- [8] Cao, Y., W. Li, and J. Zhang. Real-time traffic information collecting and monitoring system based on the internet of things. in *Pervasive Computing and Applications (ICPCA)*, 2011 6th International Conference on. 2011. IEEE.
- [9] Chen, B., H.H. Cheng, and J. Palen, Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Transportation Research Part C: Emerging Technologies*, 2009. 17(1): p. 1-10.
- [10] Domingo, M.C., An overview of the Internet of Things for people with disabilities. *Journal of Network and Computer Applications*, 2012. 35(2): p. 584-596.
- [11] Fortino, G., et al. Integration of agent-based and cloud computing for the smart objects-oriented iot. in *Computer Supported Cooperative Work in Design (CSCWD)*, Proceedings of the 2014 IEEE 18th International Conference on. 2014. IEEE.
- [12] Goumopoulos, C., A. Kameas, and P. Hellas, Smart objects as components of ubicomp applications. 2009.
- [13] Johnson, W.W., et al., Development and demonstration of a prototype free flight cockpit display of traffic information. 1997.
- [14] Karthikeyan, T. and S. Sujatha, Optimization of traffic System using TCL Algorithm through FMSA and IMAC Agents. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Voulme, 2012. 1.
- [15] Katiyar, V., P. Kumar, and N. Chand, An intelligent transportation systems architecture using wireless sensor networks. *International Journal of Computer Applications*, 2011. 14(2): p. 22-26.
- [16] Kawsar, F., A document-based framework for user centric smart object systems. PhD in Computer Science, Waseda University, Japan, 2009.
- [17] Miorandi, D., et al., Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 2012. 10(7): p. 1497-1516.
- [18] Nakamiti, G., et al. An agent-based simulation system for traffic control in the Brazilian Intelligent Cities Project context. in *Proceedings of the 2012 Symposium on Agent Directed Simulation*. 2012. Society for Computer Simulation International.
- [19] Pang, M.-b. and X.-p. Zhao. Traffic flow prediction of chaos time series by using subtractive clustering for fuzzy neural network modeling. in *Intelligent Information Technology Application*, 2008. IITA'08. Second International Symposium on. 2008. IEEE.
- [20] Pyykönen, P., et al. Iot for intelligent traffic system. in *Intelligent Computer Communication and Processing (ICCP)*, 2013 IEEE International Conference on. 2013. IEEE.
- [21] Rodríguez-Molina, J., et al., Combining wireless sensor networks and semantic middleware for an internet of things-based sportsman/woman monitoring application. *Sensors*, 2013. 13(2): p. 1787-1835.
- [22] Sanchez, T., et al., Adding sense to the internet of things—an architecture framework for smart object systems. *Pers Ubiquitous Comput*, 2012. 16(3): p. 291-308.
- [23] Spiess, P., et al. SOA-based integration of the internet of things in enterprise services. in *Web Services*, 2009. ICWS 2009. IEEE International Conference on. 2009. IEEE.
- [24] Yin, Y. and D. Jiang. Research and application on intelligent parking solution based on internet of things. in *Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2013 5th International Conference on. 2013. IEEE.
- [25] Yu, X., F. Sun, and X. Cheng. Intelligent urban traffic management system based on cloud computing and Internet of Things. in *Computer Science & Service System (CSSS)*, 2012 International Conference on. 2012. IEEE.