GSM based Vehicle-to-Vehicle Communication using Multi-Agent Intelligent System

Ayman M Mansour
Department of Communication, Electronics and Computer Engineering
Tafila Technical University
Is, Tafila
Jordan
mansour@ttu.edu.jo

Abstract—Communication between vehicles is highly desirable especially in the recent trend of auto driving cars and carelessness of drivers. Current methods largely rely on the awareness of the driver to his driving environment. In this paper, we have developed a multi-agent system approach for Vehicle to Vehicle communication. A multi-agent system is formed by a community of agents that exchange information and proactively help one another to achieve the goals set by the system designer to achieve the best driving behavior. We show how agents, equipped with decision rules can collaborate to monitor the behavior of driver. Using the popular agent language JADE [8, 10], we have constructed a small group of agents and generated simulated results.

Keywords—Fuzzy Logic, Inference system, Vehicle to Vehicle Communication, Multi-agent system

1 Introduction

1.1 Problem Statement

IN this paper we designed and developed a multi agent system for vehicle to vehicle communication The developed multi-agents system will provide agents that are capable of collaborating with one another, sharing vehicle and driver information and exchanging knowledge in a proactive way (Figure 1). The agent is a special software working for its human client/clients to perform certain tasks that imitate human agents or systems and it has the ability to be autonomous in its action. This collaborative system will enhance driving behavior and reduce accident comparing to the existing system.

Agent in the developed system is equipped with a decision engine, which will enable it getting some important cues that can be used in early detection of an abnormal driver behavior problem. Multi-agent systems have been widely adopted in many application domains because of its offered advantages [5,6]. In multi agent system the complex task may be divided to different part and each part can be handled concurrently by specific agent. Different tasks or services can be distributed among the agents based on their complexity so each agent will deal with certain number of task rather than dealing with all tasks.

In the proposed system, any information presented to any agent in the network or any decision rule added to the agent decision system will be shared with other agents not necessarily in the same network. The used reasoning is based on fuzzy logic [1] involving fuzzy rules and fuzzy reasoning implemented using the freeware FuzzyJess [2]. Integrating a decision mechanism into agents make them more proactive and encourage closer agent-human collaboration [3], [4].

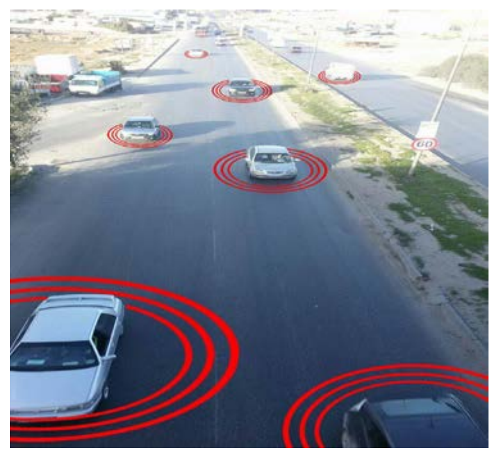

Fig. 1. Vehicle to Vehicle communication.

The developed system relies on SMS messages because of the wide availability of GSM network comparing to other network. The software provides the capability of knowledge sharing between different vehicles. To implement the proposed system we had many possible techniques. One of these is to use the peer-to-peer (P2P) model, where all the peers are treated equally and the service is distributed among all the peers of the network. However, for the pure P2P model, it is difficult to maintain the coherence of the network, discover new peers and ensure security. Also security is a quite important issue since each node is allowed to join the network without any control mechanism. We thus take the hybrid P2P model, a model between the Client/Server model and the pure P2P model using GSM technology through SMS messages. This model satisfies our need. This system provides quick response rate and the decision is faster than the manual methods. Vehicle-to-vehicle (V2V) communication vehicle can be a mesh network, meaning every node (car, smart traffic signal, etc.) could send, capture and retransmit signals.

Through Vehicle-to-vehicle (V2V) communication vehicle can send messages to each other with information about what they're doing. This data would include speed, location, direction of travel, braking, and loss of stability. Vehicle-to-vehicle can also communication using short-range communications. Vehicle-to-vehicle (V2V) communication vehicle will provide enough time for even the most distracted driver to take his foot off the gas in order to avoid accident. On the first cars, warnings might come to the driver as an alert, or as a red light that flashes in the instrument panel.

1.2 Architectures of the Agents

In this paper, we will focus on Car Monitoring Agents (CMAs). A CMA can communicate with other CMAs in the same or different driving zones. Fig. 2 shows the general structure of a CMA and how it is being connected to other agents. Within each zone, there would be many CMAs.

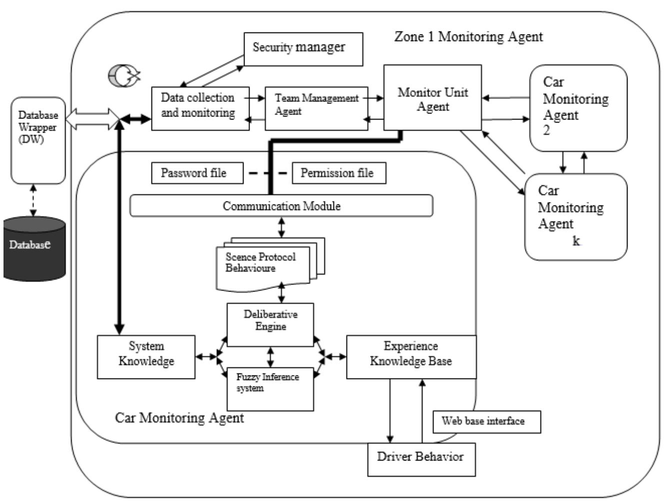

Fig. 2. General architecture of a CMA

Functionally speaking, the CMA's architecture consists of six components:

- **Communication module**. It deals with the high-level communication details, managing all the incoming and outgoing messages.
- **Domain specific knowledge base**. It keeps a set of experiences that are elicited from domain experts. It represents the agent's view of the world and maintains the variables and criteria that the Fuzzy Inference Engine can use to infer possible events (e.g., Driving Risk alarm) on a particular vehicle.
- **System knowledge base**. It contains the specifications of the scenes and the performative structure.
- **Fuzzy Inference Engine**. This decision engine contains the reasoning algorithm, which is the agent's brain. It performs reasoning tasks and executes local events.
- **A Set of scene protocol behavior models**. They model current state of the agent in a scene. There is a scene protocol behavior for each clone of the agent.
- **Deliberative engine**. Using the domain specific knowledge, system knowledge, fuzzy inference engine and the current state of the agent represented in the scene protocol behaviors, this engine makes decision on the next action by taking into account the restrictions imposed by the system designer.

1.3 Application of communication between Vehicles

There are many applications of communication between vehicles. This includes the following:

1- Measuring the distance between the two cars. This helps driver avoid collision. This option can be used if the driver is unable to determine the distance between his vehicle and the vehicle in front due to weather conditions, weakness of driver seeing or car's lights not sufficient.

2- Data transferring between the two cars (video, audio, pictures) after getting the approval from the two vehicles.

3- Blood pressure measuring of the driver through placing a sensor on the steering wheel. If blood pressure is exceeding certain limits, the agent will send an alert to the nearest car or nearest hospital. The alert will be provided to the driver himself to stop driving the car immediately.

4- Send an alert to the nearest police center or hospital in case of an accident by placing sensor on the airbag using GPS technology to capture the accident location.

5- A sensor to measure the inside temperature of car and send an alert to the driver in the event of high or low temperature, which exceeds normal ranges. This is useful for children being in the car that cannot handle the low temperature or high temperature.

6- If an error or a damage suddenly accrued in the car preventing it from moving, the agent sends an alert to the closest car in order to get a help. The location of the not working car will be provided to other vehicles.

7- Monitoring and tracking the movements of the driver eyes and always check the angle of the driver face. An alarm will be initiated (loud volume) if the driver is sleeping or his eyes were plinking or he is looking in another direction far away from the road for a certain time. This option can be used for long distance driving.

8- In the case the approaching car is coming in extra speed and the distance between the two cares is becoming small, an emergency flashlight will be activated.

9. A suddenly brake alarm will be provided to the following cars to avoid collisions.

The CMA is equipped with the rule-based reasoning capability. More specifically, we have designed a "Fuzzy driving Assistant System". Fuzzy logic is a well-established methodology that is effective for systematic handling of deterministic uncertainty and subjective information. Fuzzy logic is useful especially when a mathematical model of a system is not available and rules of the thumb from domain experts are available. It has been successfully used to solve challenging industrial and medical problems in practice, some of which are very difficult to solve without it. The Assistant system is a common rule-based system that uses fuzzy sets theory.

The reasoning is based on fuzzy logic. The structure of the Assistant System includes four components: Fuzzifier, Inference Engine, Knowledge Base, and Defuzzifier. The Fuzzifier translates crisp inputs into fuzzy values. The Inference Engine is the part that controls the process of deriving conclusions. It applies a fuzzy reasoning mechanism to obtain a fuzzy output using rules and the fuzzy values. The Knowledge Base contains a set of fuzzy IF-THEN rules and a set of membership functions of fuzzy sets. These rules represent the knowledge that the CMA possesses. The Defuzzifier coverts the fuzzy output into a crisp value that best represents the out fuzzy set. The Defuzzifier uses the center of gravity scheme. The implication methods used in the proposed system are min (minimum), which truncates the individual output fuzzy sets, and max (maximum), which scales the resulted output fuzzy sets. The input to the implication process is a single number given by antecedent.

The driving risk-monitoring task is performed by the Inference Engine that evaluates all the rules in the rule base and combines the weighted consequents of all relevant rules into a single output fuzzy set. That set is then defuzzified to produce a crisp similarity value.

The “Fuzzy Driving Assistant System” is experience-based as experience plays a key role in the design of it. The decision rules are used to build the Knowledge Base. The Assistant system uses a number of parameters related to driving environment.

The input and output variables will be defined in order to be used by the Fuzzy Inference Engine, and each variable is fuzzified by input fuzzy sets. The fuzzy sets used in fuzzifying the Input and Output variables are shown in Table I. Bell fuzzy sets are specified by three parameters a, b and c while the gaussian fuzzy set is specified by two parameters a and b and trapezoidal fuzzy set is specified by four parameters a, b, c, and d.

TABLE I
DEFINITIONS OF FUZZY SETS USED IN THE PROPOSED SYSTEM

Fuzzy Set Type	Fuzzy Set Definition
Trapezoidal	$\mu_{Trap}(x) = \begin{cases} 0, & a < x \\ -\frac{1}{b-a}(a-x), & a \le x \le b \\ \frac{1}{d-c}(d-x), & c \le x \le d \\ 0, & d < x \end{cases}$
Bell	$\mu_B(x) = \frac{1}{1+\left\lvert\frac{a-x}{b}\right\rvert^{2c}} \quad , \quad c > 0$
Gaussian	$\mu_G(x) = e^{-\frac{(a-x)^2}{2b^2}}$

Due to the space limit, we will only provide part of the rules – those involved driving risks behavior (Figure 3). The distance between two vehicles, the speed of vehicles and the change of steering angles will be the input to the system and the Driving Risk Value will be the output. Both the input and the output are fuzzy variables.

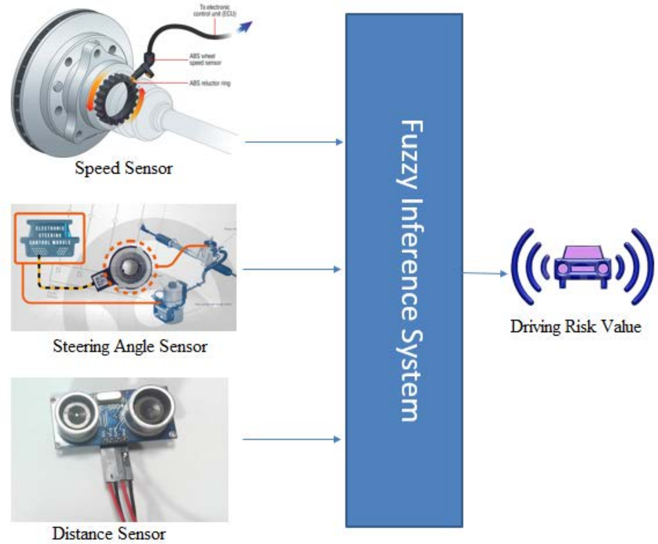

Fig. 3. Fuzzy Inference System of Driving Risk Value

There are five fuzzy sets for the input variable Distance (Fig. 4), five fuzzy sets for the input variable speed (Fig. 5), three fuzzy sets for the input change of steering angle (Fig. 6) and four fuzzy sets for the output variable Driving Risk Value (Fig. 7).

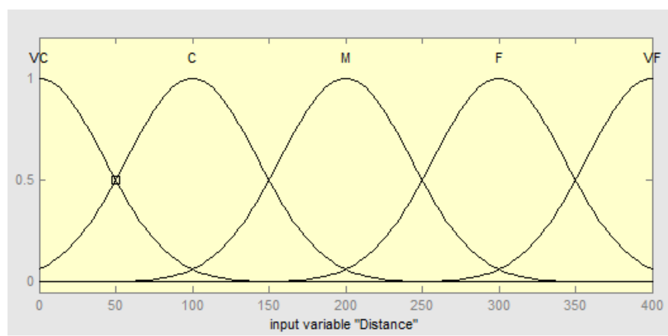

Fig. 4. Fuzzy sets for Distance (meter)

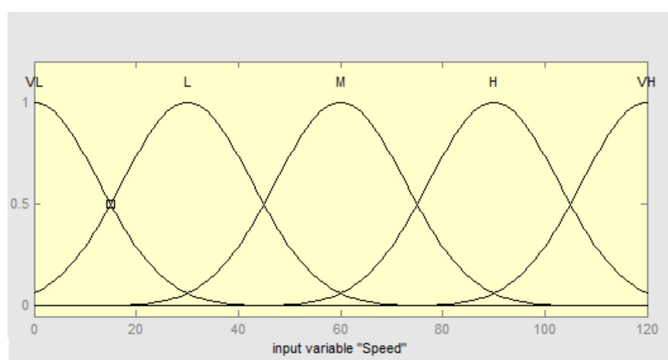

Fig. 5. Fuzzy sets for Speed (m/s)

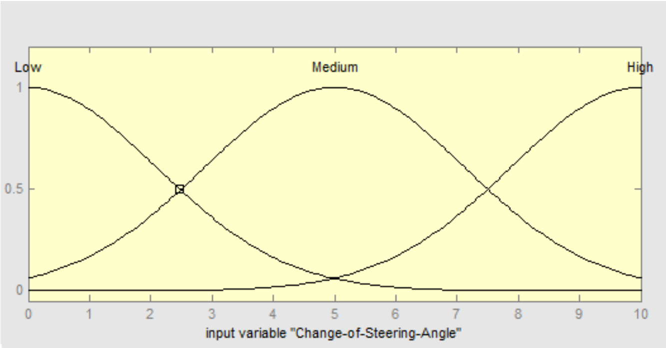

Fig. 6. Fuzzy sets for Change of Steering Angle

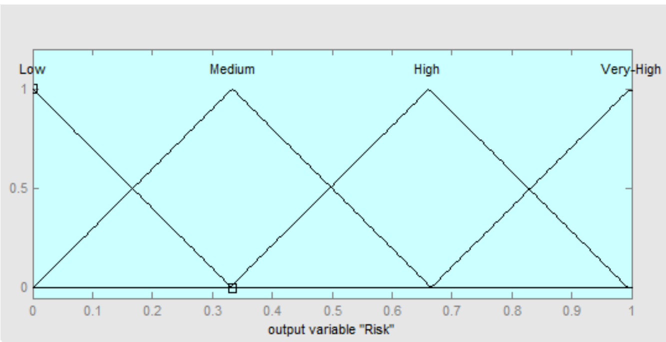

Fig. 7. Fuzzy sets for Driving Risk Value

A sample of the rules used to determine Degree of Driver Risk Value:

- If Distance is *Very Close*, Speed is *Low* and Change of Steering Angle *is Low* the Driving Risk Value is *Medium.*
- If Distance is *Medium*, Speed *is Medium* and Change of Steering Angle *is High* the Driving Risk Value is *High.*

The Degree of Risk will be a value between 0 and 1. A higher score represents a higher risk. The defuzzified output generated by the fuzzy Inference System will be categorized to the following levels:

• Level 1: "Degree of Risk" score from 0.00 to 0.50 represents Low Risk.

• Level 2: "Degree of Risk" score from 0.50 1.00 represents High Risk.

Those levels are decided by the domain experts. In this design, we use the following settings: the Min-Max fuzzy inference and the centroid defuzzifier.

2 System Implementation

For system construction and execution, JADE 3.7 agent platform (Java Agent DEvelopment Framework) is adopted [2]-[4]. JADE is an open-source software framework. JADE is a widely used package in multi agent system implantation [3]. JADE is following the specifications laid by The Foundation for Intelligent Physical Agents (FIPA) for multi agent system implementations [1]. It provides a set of Java classes that makes it easy to implement the systems. JADE can run on a variety of operating systems including Windows and Linux. The JADE platform includes most of agent's specifications. Each agent is implemented in JADE as a single thread. JADE provide a multi thread environment that allows the agents to execute parallel tasks. Different cooperative behaviors can effectively schedules in JADE. JADE incorporates some ready to use behaviors that commonly used by agents during performing certain task. Among the others, JADE offers a behavior that allows full integration with JESS, which is a rule, based engine that performs all the necessary reasoning.

Since agents are normally distributed at different location (Figure 8). The developed system includes inter-vehicle communication to transmit traffic related information to a group of Vehicles. The system is also provided with vehicle-to-roadside communication where the roadside unit sends messages to all Vehicles in the cluster. Each location will have a runtime environment that hosts the agents, which is called a "Container". Each container has AMS and DF agents. The containers will communicate with each other using a pre-defined protocol. One container needs to be assign as main container, which represents the bootstrap point of a platform. It is the first container to be launched and all the other containers must join it by registering with it. For my system, the main container hosts 5 CMA. JADE also provides tools that manage both locally and remotely agent life cycles including create, suspend, resume, freeze, thaw, migrate, clone and kill.

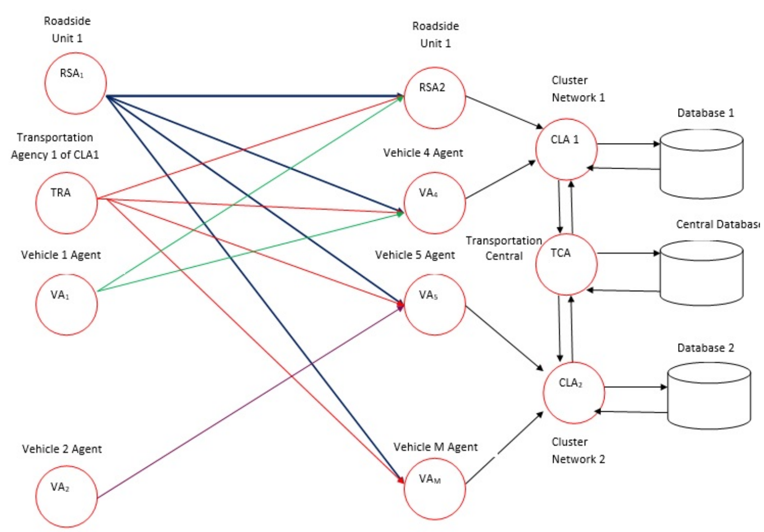

Fig. 8. The Developed Vehicle to Vehicle Multi-Agent system

Agents and Roles

Each agent adopts some role, that is, a pattern of behavior (tasks). The illocutionary actions performed by an agent are constrained according to the role it is playing. In a Vehicle to vehicle communication, agents play the roles of cars, buses, trains, traffic lights, and so on [4].

Dialogical Framework

The concepts involved in the communication among agents have to be fixed. Agents interact through illocutions or speech acts. The proposed system has to establish the acceptable illocutions defining the ontology, the common language to represent the world, and the common language for communication and knowledge representation. The dialogical framework is composed by ontology, a representational language for the concrete domain, a set of illocutionary particles, and a communication language. The representation language for the domain is first-order logic. The set of illocutionary particles used are: query, inform, request, offer, accept, withdraw, and refuse.

Scenes

Interaction between agents occurs within a scene. A scene is basically a group meeting, and it is composed of a set of agents playing different roles and communicating with a well-defined communication protocol. Our system is composed by several scenes connected among them. To specify a scene, the first step is to identify which are the roles that will participate in the scene. Agents with these roles can enter and exit the scene maybe to go to another one. The second step is to define the communication protocol [4]. The communication protocol is fixed and it is impeded in JADE. The scene is structured as a graph where the nodes represent the different states of a conversation and the arcs are labeled with expressions of the communication language. It is mandatory an initial state and a final state. An agent enters or exits to and from different states; this is represented by a little white triangle or a little black triangle labeled with the CMAs' names.

Interaction between agents occurs within a scene. Our system is composed by several scenes. We consider the protocol of a scene to be the specification of the possible dialogues the participating agents have. A scene can be described as a graph, which can be regarded as a state diagram. The nodes represent different states of a conversation. The graph has a single initial state and a set of final states representing different endings of the conversation. Fig. 9 shows the state diagram of a scene where

$Q = \{X0 - X4\}$ – a set of states of the automaton:

X0 – initial state;

X1 – waiting for a response to the request;

X2 – checking the data;

X3 – lack of communication;

X4 – data processing,

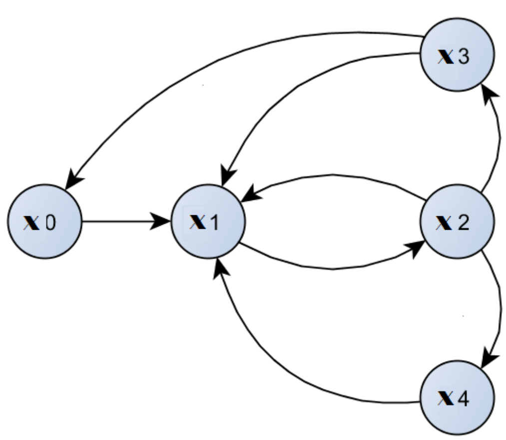

Fig. 9. The state diagram of a scene in multi agent environment.

Fig. 10 shows a conversation between the two agents in the proposed system using the Petri nets approach. The interesting thing about this approach is that we can describe more precisely what is happening during a conversation between agents [3]. A Petri net is defined as a graph containing two sorts of nodes: *places and transitions*. This graph is built in such a way that the arcs can only link places to transitions or transitions to places. Places are graphically represented by circles and transitions by bars. The places correspond to the state of the agent during a conversation stage.

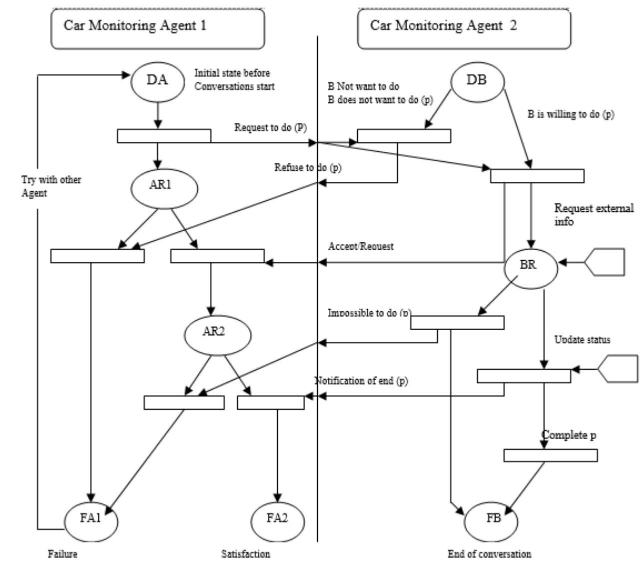

Fig. 10. Conversational model of a medical Decision task between two PAs using the Petri nets approach [9]

The proposed way of sending the data to the database is not only using regular internet networks but also using GSM technology through SMS messages. Using SMS messages is very helpful in rural areas where the internet connection is not available. The SMS architecture is shown in Figure 11. The SMS message will be transmitted from the agent GSM module to nearest Base Transceiver Station (BTS) through a wireless Channel [8,9,10]. The SMS will be received and passed to Base Station Controller (BSC). The BSC handles the operation of the BTS. The message will leave BSC and move to the Mobile Switching Center (MSC). The Mobile Switching Center acts like a switching device which switch data between users on the network based on routing information provided by the Home Location Register (HLR). The HLR contains relevant data about network subscribers: their status, location and thus its routing information. The message will be stored within Short Message Service Center (SMSC) which will forward the message when the receiving Agent becomes available. The SMSC is software that manages the processes of queuing the messages, billing the sender and returning receipts if necessary. The Subscriber details will be validated by Visitor Location Register (VLR) and Equipment Identity Register (EIR).

VLR will verify that the message transfer does not violate the supplementary services invoked or the restrictions imposed. The visitor location register is a database that contains temporary information about subscribers homed in one Home Location Register (HLR) who are roaming into another HLR. EIR is a database to determine if the service of a GSM mobile Subscriber is authorized, unauthorized, or if it should be monitored. It uses International Mobile Equipment Identity (IMEI) to identify each Subscriber device. An IMEI is considered as invalid if it has been reported stolen or is not type approved.

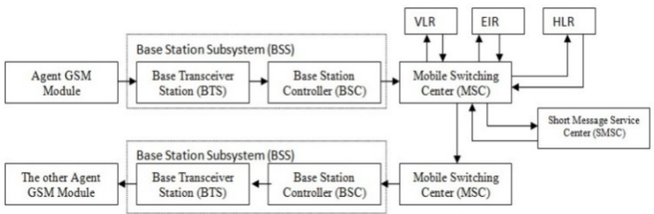

Fig.11. SMS Architecture.

MSC asks HLR for reception location. The MSC will deliver the message to the specific mobile subscriber through the proper base station. The message will be released and moved to the appropriate MSC and then forwarded to nearest Base Station Subsystem (BSS) and it will be received to its destination agent. The used GSM Module (Figure 12) is Mini GSM / GPRS breakout board is based on SIM800L module, supports quad-band GSM/GPRS network, available for GPRS and SMS message data remote transmission. The board features compact size and low current consumption. With power saving technique, the current consumption is as low as 1mA in sleep mode. The GSM modules uses AT commands. AT commands is used to collect information by controlling GSM Modems. A GSM modem is a special device that uses a SIM card, and it operates like mobile phones. AT is the abbreviation of ATtention.

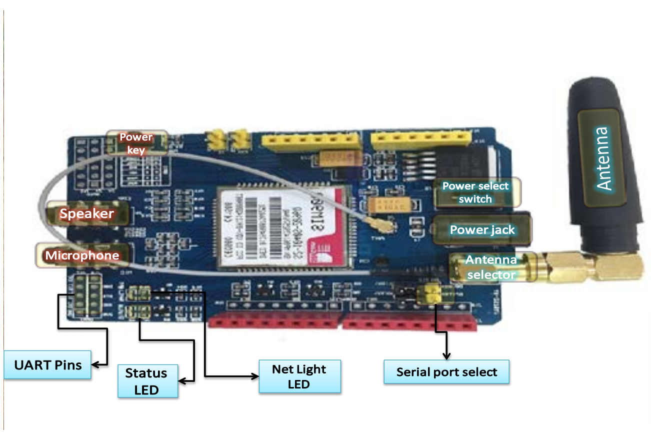

Fig12. SIM908 GSM module

GSM module requires one SIM card. This module is capable to accept any network SIM card. The GSM module has a unique identity number like mobile phones have. The GSM module is used to send an SMS to the user‘s cell phone number. When the vehicle data has been entered and the send order has been given, a signal to the GSM module which then sends a message to the physician user that is predefined by the programmer. GSM SIM 900 Quad-band GSM/GPRS engine, works on frequencies 850MHz, 900MHz, 1800MHz. It is very compact in size and designed with RS 232 level converter circuitry, which allows you to directly interface PC Serial port. GSM uses a combination of Time Division Multiplexing and Frequency Division Multiplexing. The baud rate can be configurable from 9600-115200 through AT command. This GSM/GPRS RS232 Module is having internal TCP/IP stack to enable you to connect with internet via GPRS. Using this module, we will be able to send & read SMS, Connect to internet via GPRS through simple AT commands. The suitable operating voltage level is 5V-12V DC.

During the development of the proposed vehicle to vehicle communication system with JADE, the following types of Java code classes are created and implemented:

Reasoning classes, which are used for the implementation of the various reasoning models and Fuzzy Inference Engine employed by CMAs, for example, the Fuzzy. In this application FuzzyJess Toolkit from the National Research Council of Canada's Institute for Information Technology has been used. It is a set of Java classes that provide the capability for handling fuzzy concepts and reasoning [2]. It is compatible with JADE. It allows the user to use Java language to define membership functions, set antecedent and consequent of a fuzzy rule, and makes a fuzzy inference. FuzzyJess uses Jess (Java Expert System Shell). Jess provides the basic elements of an expert system including fact-list, knowledge base that contain all the rules and inference engine which controls overall execution of rules.

Graphical User Interface (GUI) classes, which provide the user with a graphical interface to the multi-agent system, initiate a search, and show the results of a query to the user. This part will be developed in our future work and will be integrated with GSM/GPRS Modem to immediately transmit the vehicle data.

*White Board classes***,** JADE provides the classes used to help an agent publish and search for services through method calls. These classes help agents use the White Board.

A prototype design for proposed System is successfully developed. A PIC microcomputer is a processor with built-in memory and RAM. PIC16F877 is one of the most commonly used microcontrollers especially in automotive, industrial and home applications. This prototype design is responsible for receiving data from vehicle sensor.

Fig. 13 shows the Transmitter Prototype kit and Fig. 14 shows the Receiver kit Prototype.

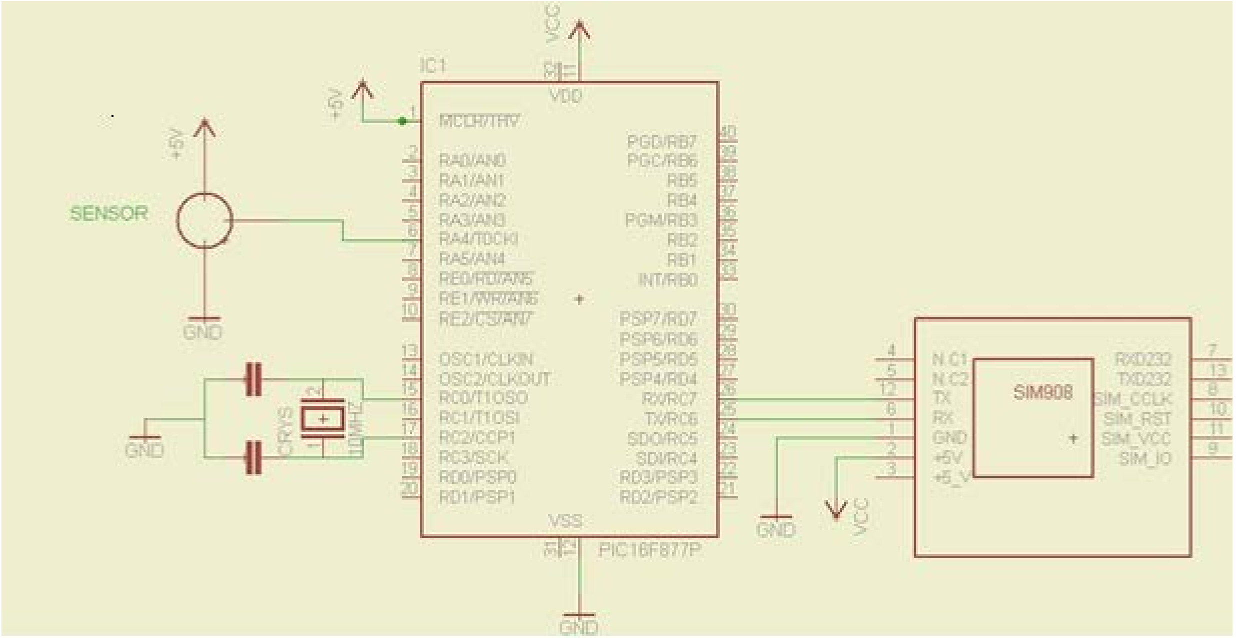

Fig. 13. Schematic of transmitter prototype kit circuit

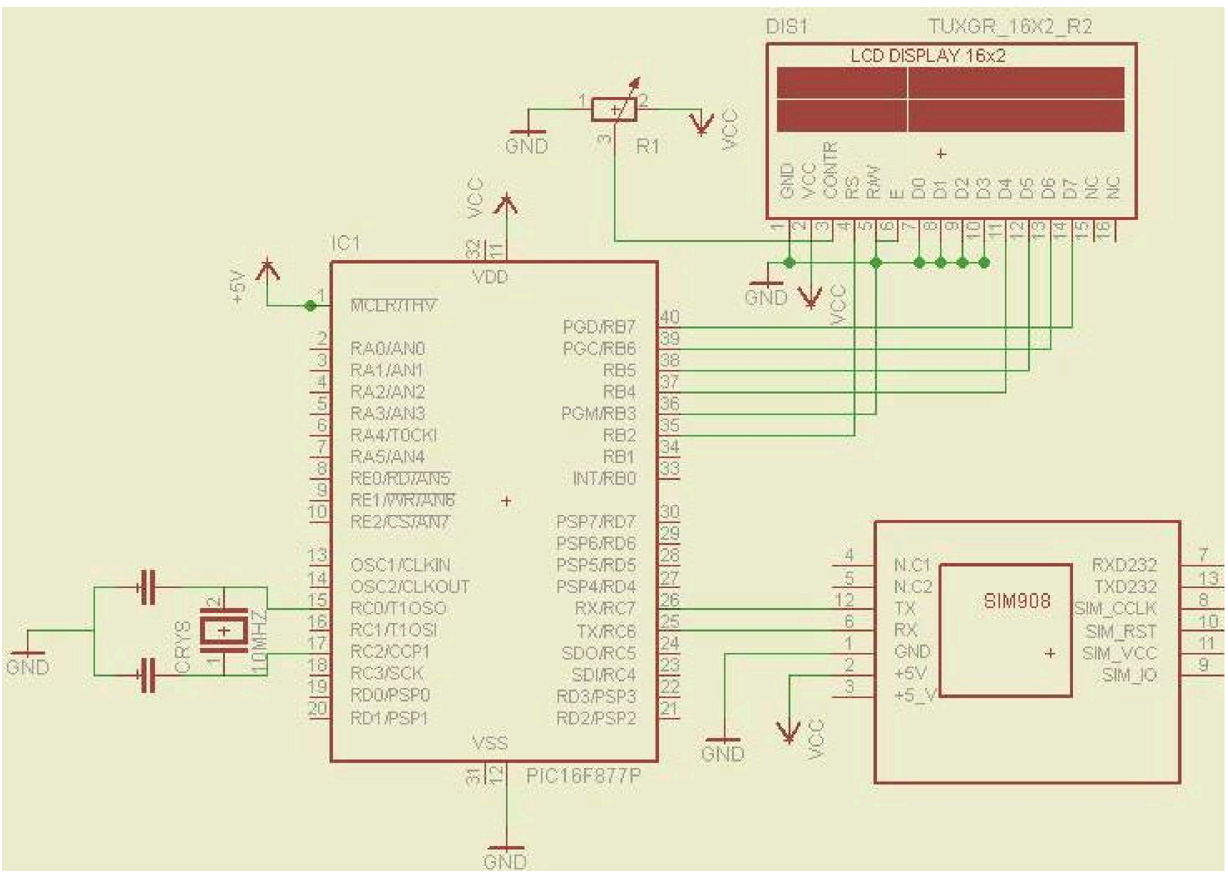

Fig. 14. Schematic of receiver prototype kit Circuit

The vehicle transmition kit shown in Fig. 15 consists of four sensors and the associated circuitry to get the data. There are mainly four units in transmition kit: sensor unit, data collecting unit, microcontroller unit and GSM module unit. The output of the sensors unit is connected to a data-collecting unit, which is connected to the microcontroller. The Fuzzy Inference Engine in a microcontroller will use the gathered data and check if the "Driving Risk Value" is within safe levels. If the "Driving Risk Value" is High Risk, the microcontroller immediately triggers the Buzzer Alert Unit to activate an alarm; and LED Display Unit starts flashing. At the same time, the Microcontroller Unit sends an alert message about the situation to other Vehicles. The data will be displayed on the LCD Display Unit. Multiple messages can also be sent to the Central Transportation Agency, Civil defense unit or Police station with GPS location of the Vehicle.

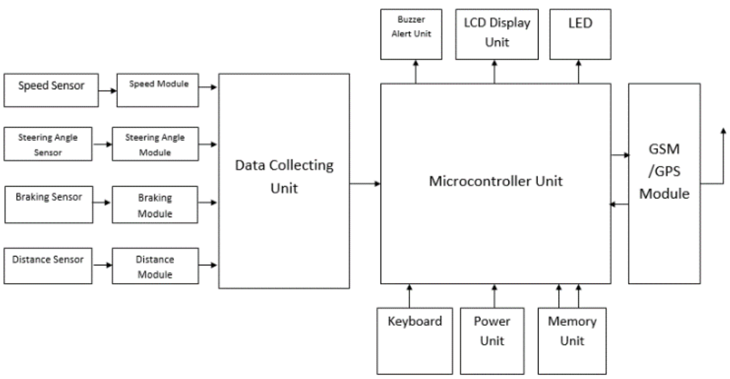

Fig. 15. The structure of Vehicle Transmition Kit module

The functionality of vehicle receiving kit is divided into three main steps as shown in Figure 16.

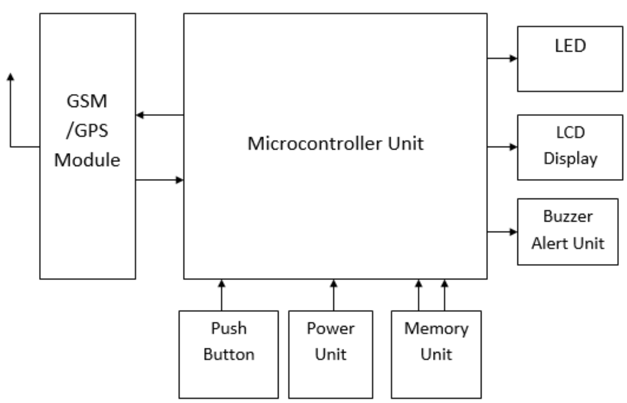

Fig. 16. Block diagram of Vehicle Receiving Kit Module

In the initial step, the Vehicle Receiving Kit receives the transmitted data with the initial decision made by its VA. After that, Microcontroller Unit sends an activation signal to other units attached with it such as Buzzer Alert Unit, LEDs Unit and LCD Display Unit. In the last step, if the Driving Risk Value is High, buzzer will be activated, LED blinks and LCD shows the other vehicle data. The decision is based on the most reliable rules in the whole muli-agent system. The system can be accessed from the Transportation central unit through GUI as shown in Figure 17. The Logging and Security Unit allows only the authorized people to get into the system.

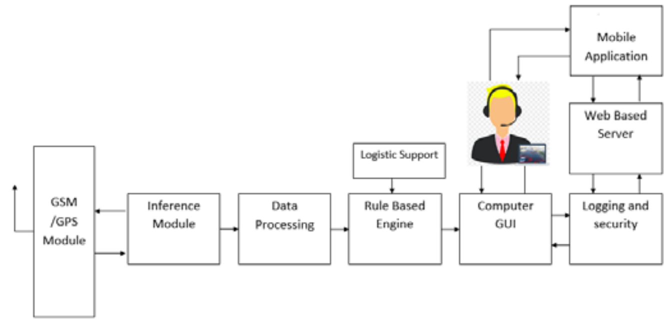

Fig. 17. Block diagram of Central Transportation Agent (CTA) module

3 Experiment results

We carried out some experiments to evaluate the proposed system in JADE and FuzzyJess. More specifically, we sought to assess the agents ability to communicate with each others. We form the agent system by implementing five CMAs. Network Simulator 2 (ns2), is computer simulation program that is used to evaluate the performance of the communication networks.

NS2 is free & open source publicly supported simulation tool that used in this paper, to evaluate the performance of some network topology examples. NS is an event driven network simulator developed at UC Berkeley that simulates variety of IP networks. It implements network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations. NS2 written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT).

In this experiment, data communication network has been built and tested with different possible topologies to evaluate the performance of each one. This network connects five nodes that represent five different vehicles that connected with each other. Every one of these vehicles sends & receives data from the other. Each link in this network has significant information determine the capacity, delay, type of data & the interval time of sending data, but all vehicles in the network have the same packet size. The performance of these topologies can be indicated by: the network wide mean packet delay, total cost network and set the advantages & disadvantages of each one.

In this experiment, we have fixed length packet size, which equal to 2500 bits. Note that the packet type is Poisson, so we used the exponential traffic source to generate data. The intervals of sending packets of each node:-

This calculated from the amount of traffic generated from each vehicle. Table 2 shows the intervals of sending one packet.

TABLE II: INTERVALS OF SENDING ONE PACKET

Source/ Destination	Car (1)	Car (2)	Car (3)	Car (4)	Car (5)
Car(1)	------	1.44	1.84	0.42	0.44
Car (2)	1	-----	7.45	1.34	1.44
Car (3)	1	0.5	-------	1.22	1.26
Car (4)	0.45	1	0.5	--------	0.18
Car (5)	0.5	0.5	1	1	-----

- Delay of sending packets:-

This is the time of sending the packets from one vehicle to other one. From the distance given in table 2 in the outline and speed electromagnetic signal, we can calculate the delays of each one. Table 3 shows the delay of sending packets in unit of millisecond.

TABLE III: DELAY OF SENDING PACKETS IN UNIT OF MILLISECOND

Source/Destination	Car (1)	Car (2)	Car (3)	Car (4)	Car (5)
Car(1)	-----	9.6	12.6	16.4	19.3
Car (2)	7.2	------	8.3	9.7	12.6
Car (3)	14.6	8.3	-------	8.7	13.3
Car (4)	19.3	12.6	13.3	5.2	-----
Car (5)	19.3	14.6	13.3	6.6	-----

- Wide mean packet delay:-

To calculate this value, we must run the program for enough time. The delay has been evaluated to 100000 packets using the following topologies (Figure 18, Figure 19 and Figure 20):

❖ Ring Topology:-

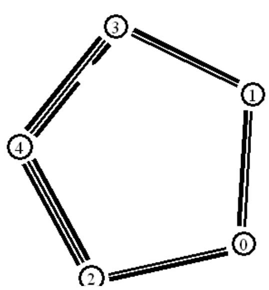

Fig. 18. Ring Topology

Delay = 212.3ms

❖ Star Topology:-

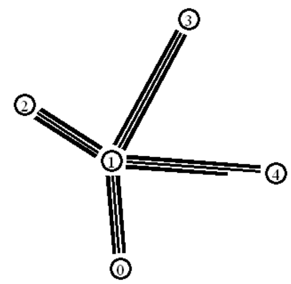

Fig. 19. Star Topology

Delay = 312.7 ms

❖ Mesh Topology:-

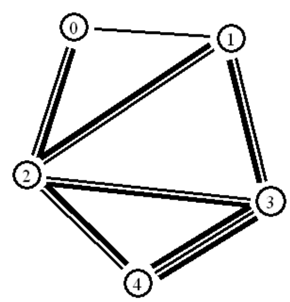

Fig. 20. Mesh Topology

Delay = 239.3 ms

According to the wide delay time, we see that the ring & the mesh topologies approximately have very close delay time, but the star topology has delay time greater that the other. If we have any link failed between any two nodes in the ring topology (Figure 21), the system will have another path that will be used to reach the destination. Such option is not available in the star topology (Figure 22). Mesh topology (Figure 23) has the same behavior as the ring topology if any link failed happened. In star topology if the center node failed, the system will be shutdown. Considering all of above, mesh topology has the best performance than other.

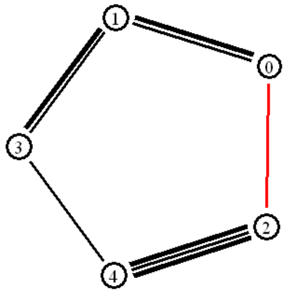

Fig. 21. Ring with Link Failed

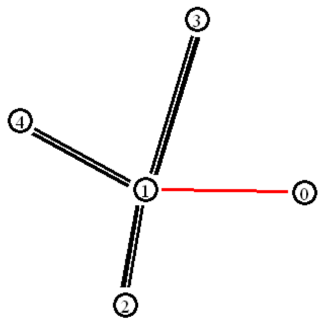

Fig. 22. Star with Link Failed

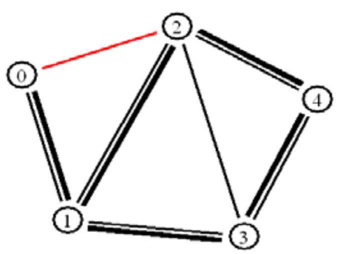

Fig. 23. Mesh with Link Failed

4 Conclusion

In this paper, we have proposed a multi-agent system for Vehicle-to-Vehicle communication. As a first step, we have designed a multi agent system and started the implementation using JADE and FuzzyJess software packages. As a next step, we will use real data gathered from online sensors placed on vehicles using embedded system with GSM modem to fully assess the performance of the Multi-Agent system.

References

[1] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, pp. 338-353, 1965.

[2] R. Orchard, "Fuzzy Reasoning in Jess: The FuzzyJ Toolkit and FuzzyJess," Third International Conference on Enterprise Information Systems, Setubal, Portugal, pp. 533-542, 2001.

[3] Jacques Ferber, Multi-Agent Systems: Intro. to Distributed System. Addison-Wesley, 1999.

[4] Hong Lin, Architectural Design of Multi-Agent Systems: Technologies and Techniques. Information Science Reference press (an imprint of IGI Global), 2007.

[5] FIPA The foundation for intelligent physical agents. See http://www.fipa.org.

[6] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood, Developing Multi-Agent Systems with JADE. Wiley, 2007.

[7] K. Chmiel, et al, "Testing the Efficiency of JADE Agent Platform," Proceedings of the 3rd International Symposium on Parallel and Distributed Computing, Cork, Ireland, IEEE Computer Society Press, Los Alamitos, CA, pp. 49-57, 2000.

[8] Bellifemine, F., G. Caire, T. Trucco, and G. Rimassa., Jade Programmer's Guide. JADE 3.7, 2003. See http://jade.cselt.it.

[9] Ayman Mansour, Hao Ying, Peter Dews, Yanqing Ji, Margo Farber, John Yen, Richard E. Miller, and R. Michael Massanari, "A Multi-Agent System for Detecting Adverse Drug Reactions," Proceedings of the 29th NAFIPS, Toronto, ON, Canada, July 12-14, 2010.

Ayman M Mansour received his Ph.D. degree in Electrical Engineering from Wayne State University in 2012. Dr. Mansour received his M.Sc degree in Electrical Engineering from University of Jordan, Jordan, in 2006 and his B.Sc degree in Electrical and Electronics Engineering from University of Sharjah, UAE, in 2004. He graduated top of his class in both Bachelor and Master. Currently, Dr. Mansour is an associate professor in the Department of Electrical Engineering, Tafila Technical University, Jordan. His areas of research include communication systems, multi-agent systems, fuzzy systems, data mining and intelligent systems. He conducted several researches in his area of interest. Dr. Mansour is a member of IEEE, Michigan Society of Professional Engineers, IEEE Honor Society (HKN), Society of Automotive Engineers (SAE), Tau Beta Pi Honor Society, Sigma Xi and Golden Key Honor Society.