International Journal of Database Theory and Application
Vol.7, No.6 (2014), pp.1-12
http://dx.doi.org/10.14257/ijdta.2014.7.6.01

A Vehicle Monitoring System Based on STeCEQL

Huiyong Li and Yixiang Chen

Software Engineering Institute, East China Normal University, Shanghai, China
lihuiyongchina@126.com

Abstract

The vehicle monitoring system is a very important class of information systems. However, the existing monitoring systems generally require the support of the large database systems and the monitoring event could not be processed in real time. The research of internet of vehicles has proposed the novel ideas and methods to build the monitoring system. In this paper, we construct a vehicle monitoring system based on a complex event query language: STeCEQL. Firstly, we give the approach of verifying the reasonableness of the monitoring events through building the model of the vehicles. Then we propose the method of describe the vehicle monitoring events by STeCEQL. Meanwhile, we give the complex event detection algorithms based on the event tree. Finally, the simulation results demonstrate the effectiveness of the vehicle monitoring approach of internet of vehicles.

Keywords: *internet of things, mobile system, vehicle monitoring system, complex event query language, query planning*

1. Introduction

The internet of vehicles is an important achievement of the internet of things [1]. There are not only a large number of non-mobile agents in the internet of vehicles, but also a lot of fast-moving mobile agents. In the system, the various sensors carried by each agent produce a continuous flood of data: time, position, direction, speed, temperature, etc [2, 3]. However, the conventional database technology can not deal these data effectively.

The vehicle monitoring systems are generally used to monitor the operational state of the various vehicles [4, 5]. Nowadays, the vehicle monitoring systems will directly transport the large amounts of data to the database of the monitoring center. Then the monitoring results will be expressed by the trajectory tracking technology and the data mining technology. There are great pressures of transporting the amount data of vehicles through the traditional monitoring approach of vehicle monitoring system [6]. Another problem of this method is that it is not able to monitor the vehicle in real time [7].

In recent years, the complex event processing technology has been successfully applied in the internet of vehicles projects. In internet of vehicles, the large amount of raw data generated by various sensors is seen as basic events. Through the complex event query language, the system will filter these basic events into meaningful complex events. Therefore, the massive amounts of data of the system can be effectively processed in real time. Moody K proposed a complex event of query language of internet of vehicles: SpaTec. This EQL has been applied to the London's bus monitoring system [4, 5]. Jin B proposed a complex event query language of internet of vehicles: CPSL [6]. We also proposed a complex event query language: STeCEQL. And it can describe a variety of temporal and spatial attributes of the events.

In this paper, we construct a vehicle monitoring system based on the complex event query language and propose the corresponding monitoring strategy. In our vehicle monitoring system, the monitoring events were developed by the monitoring center. Then, the monitoring event will be converted to the complex event query expressions and transmitted to the embedded information system of the vehicle. In the embedded information system of the

ISSN: 2005-4270 IJDTA

vehicle, the data of the vehicle could be real-time filtered by the complex event expressions. This will greatly reduce the computational load of the monitoring center.

The remainder of this paper is organized as follows: Section 2 gives the method to verify the monitoring events through building the model of the activity of the vehicle. Section 3 gives the method to describe the vehicle monitoring events by STeCEQL language. Meanwhile proposes the event detection algorithm based the event tree. Section 4 proposes the architecture of the vehicle monitoring system and the monitoring strategies of the system. Section 5 shows the effectiveness of the vehicle monitoring strategies through the simulation results. The last Section concludes this paper.

2. The Method of Verifying the Vehicle Monitoring Events

2.1. The Formal Models of the Vehicles' Behavior

The STeC language extends the formal language CSP and CCS. It can effectively describe the temporality and the spatiality of the agent's actions or status and the spatial-temporal consistency of the intelligent system. The language consists of the action keyword α, the status keyword βand the communication mechanism of the system. The STeC language syntax as below:

$$A ::= Send_{(l,t)}^{G \to G'}(m,\delta) \mid Get_{(l,t)}^{G \to G'}(m,\delta)$$

$$B ::= \alpha_{(l,t)}^{G}(l',\delta) \mid \beta_{(l,t)}^{G}(\delta)$$

$$P ::= Stop_{(l,t)} \mid Skip_{(l,t)} \mid A \mid B \mid P;P \mid \square_{i \in I}(B_i \to P_i) \mid$$

$$P \Delta_{\delta} P \mid P \Delta (Get_{(l,t)}^{G \leftarrow G'}(m) \to Q) \mid P \parallel P$$

The communications among the agents were express by two atom communication process: Send and Get. The atom communication process Send has defined the agent G send the message m to another agent G' at location l and time t. It take δunit time to implement the communication action. Like the action Send, the atom communication process Get has defined the agent G get the message m from another agent G' at location l and time t. It takeδ unit time to implement the communication action.

2.2. The Method of Verifying the Reasonableness of the Vehicle Monitoring Events

Under the traffic rules constraints, all types of vehicles must obey certain behavioral norms. For example, the speed of the highway vehicles can not be lower than 70km/h and higher than 120km/h. In order to describe the behavior of these vehicles' specifications, we use the agent behavior specification language STeC to describe it. Here we give an example to describe the specification of a bus system.

Assume there is a bus system and the bus line is shown in Figure 1. We use the grid map to describe the geographical environment. The length of each grid side is 100 meters because the accuracy of GPS is generally 100 meters. The bus stations are Station1 to Station5.

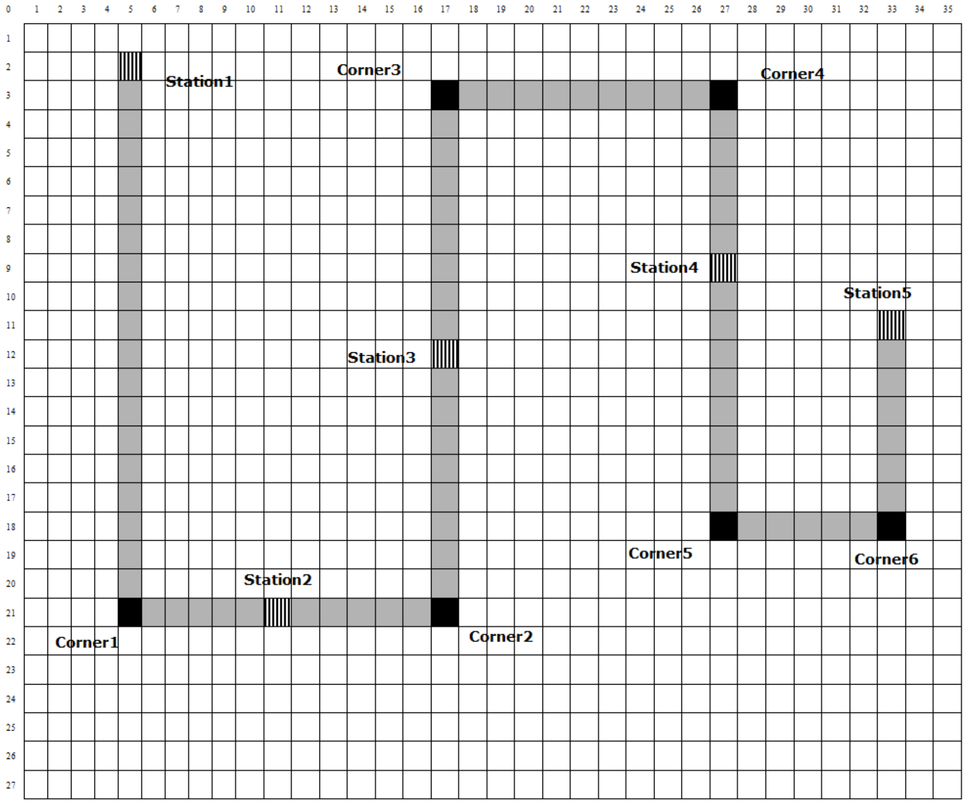

Figure 1. The Bus Line of The Bus System

Assume there is a Bus timetable as Table 1:

Table 1. A Timetable of The Bus System

Leave Station1	Arrive Station2	Leave Staion2	Arrive Station3	Leave Staion3	Arrive Station4	Leave Staion4	Arrive Station5
19:10:00	19:18:20	19:19:20	19:24:20	19:25:20	19:33:40	19:34:40	19:42:20

Let the bus name is Bus1, then the formal model of the bus is as below:
Bus1Specification1=
{

$running^{Bus1}_{(Sation1,19:10:00)}(Station2, 8min20s);$

$stop_{(Station2,1min)};$

$running^{Bus1}_{(Station2,19:19:50)}(Station3, 5min);$

$stop_{(Station3,1min)};$

$running^{Bus1}_{(Station3,19:26:20)}(Station4, 8min20s);$

$stop_{(Station4,1min)};$

$running^{Bus1}_{(Station4,19:36:10)}(Station5, 7min40s);$

}.

In order to ensure the safety of bus passengers, the bus speed should meet certain speed limits. The bus monitoring system will monitor the bus speed. Assuming the limit speed is 10m/s during the operation, then the condition can be describe as Bus1condition1=(

$v_{Bus1} \leqslant 10\ m/s$).

Here let us prove the Bus1condition1 is reasonable for the Bus1Specification1. In other words, there is some feasible solution meet the Bus1condition1.

Proof: Firstly, examine the process of the bus moving between Station1 to Staion2.

$$\because\ t1=\tau(running^{Bus1}_{(Sation1,19:10:00)}(Station2, 8min20s);) = 8min20s,$$

$$s1=\xi(running^{Bus1}_{(Sation1,19:10:00)}(Station2, 8min20s);) = 2500\ m,$$

If the bus were operated by the maximum speed limit, the distance will be

$$s1' = 10m/s \times 8min20s = 5000m.$$

$$\because\ s1' > s1$$

$\therefore$ there is a viable vehicle speed meets the Bus1condition1.

Similarly can be obtained, there are some feasible solutions to meet the Bus1condition1 in other section of the bus line.

$\therefore$ there is some feasible solution meet the Bus1condition1.

We finish the proof of this theorem. □

However, in some cases the bus speed limit is unreasonable. For example: suppose the bus limit speed is 4m/s, then the condition will be Bus1condition2=($v_{Bus1} \leqslant 4\ m/s$).

Here let us prove the Bus1condition2 is unreasonable for the Bus1Specification1. In other words, there is not feasible solution meet the Bus1condition2.

Proof: Firstly, examine the process of the bus moving between Station1 to Staion2.

$$\because\ t1=\tau(running^{Bus1}_{(Sation1,19:10:00)}(Station2, 8min20s);) = 8min20s,$$

$$s1=\xi(running^{Bus1}_{(Sation1,19:10:00)}(Station2, 8min20s);) = 2500\ m,$$

If the bus were operated by the maximum speed limit, the distance will be

$$s1' = 4m/s \times 8min20s = 2000m.$$

$$\because\ s1' < s1$$

$\therefore$ there is not a viable vehicle speed meets the Bus1condition2.

Similarly can be obtained, there are not feasible solutions to meet the Bus1condition2 in other section of the bus line.

$\therefore$ there is not feasible solution meets the Bus1condition2.

We finish the proof of this theorem. □

The above describes how to prove the reasonableness of the monitoring events. Next we describe how to express the monitoring events and process these.

3. The Method of Expressing and Processing the Monitoring Event

3.1. The Method of Expressing the Monitoring Event

The STeCEQL is complex event query language for the internet of vehicles of event driven architecture. It can express the basic events and the complex events. The syntax of STeCEQL is as follows:

ABexp:

$$attribute ::= true \mid false \mid x_a = a \mid x_a != a \mid x_a > a \mid x_a < a$$
$$\mid attribute_0 \wedge attribute_1 \mid attribute_0 \vee attribute_1$$

TBexp:

$$time ::= true \mid false \mid x_t\ BEFORE\ \ t \mid x_t\ AFTER\ \ t$$
$$\mid x_t\ EQUAL\ t \mid x_t\ OVERLAP\ t \mid x_t\ DURING\ t$$
$$\mid time_0 \vee time_1 \mid time_0 \wedge time_1$$

LBexp:

$$location ::= true \mid false \mid x_l\ EQ\ l \mid x_l\ OP\ l \mid x_l\ IN\ l$$
$$\mid x_l\ NORTH\ l \mid x_l\ SOURTH\ l \mid x_l\ EAST\ l\ \mid x_l\ WEST\ l$$
$$\mid x_l\ NORTHWEST\ l \mid x_l\ NORTHEAST\ l$$
$$\mid x_l\ SOURTHWEST\ l \mid x_l\ SOURTHEAST\ l$$
$$\mid location_0 \vee location_1 \mid locaiton_0 \wedge location_1$$

DBexp:

$$direction ::= true \mid false \mid x_d = d \mid x_d != d$$

EBexp:

$$e ::= agent^{time}(attribute_1; attribute_2; attribute_3 \cdots)$$
$$\mid agent_{location}(attribute_1; attribute_2; attribute_3 \cdots)$$
$$\mid agent^{time}_{location}(attribute_1; attribute_2; attribute_3 \cdots)$$
$$\mid agent^{time}_{(location,\ direction)}(attribute_1; attribute_2; attribute_3 \cdots)$$

CEexp:

$$e ::= e_1 \wedge e_2 \mid e_1 \vee e_2$$
$$\mid e_1\ BEFORE\ e_2 \mid \mathrm{e}_1\ AFTER\ e_2$$
$$\mid e_1\ EQUAL\ e_2 \mid e_1\ OVERLAP\ e_2 \mid e_1\ DURING\ e_2$$
$$\mid e_1\ EQ\ e_2 \mid e_1\ OP\ e_2 \mid e_1\ IN\ e_2$$
$$\mid e_1\ NORTH\ e_2 \mid e_1\ SOURTH\ e_2 \mid e_1\ EAST\ e_2\ \mid e_1\ WEST\ e_2$$
$$\mid e_1\ NORTHWEST\ e_2 \mid e_1\ NORTHEAST\ e_2$$
$$\mid e_1\ SOURTHWEST\ e_2 \mid e_1\ SOURTHEAST\ e_2$$

The following examples illustrate the usage of the STeCEQL. In the bus system of last section, if the monitoring events are: the limit bus speed is 8 m/s from Station1 to Station2; the limit bus speed is 10 m/s from Station2 to Station3; the limit bus speed is 7 m/s from Station3 to Station4; the limit bus speed is 6 m/s from Station4 to Station5. The monitoring event can be described by STeCEQL as below:

e1= $Bus1_{x_l\ IN\ Station1-Station2}(x_v > 8)$;

e2= $Bus1_{x_l\ IN\ Station2-Station3}(x_v > 10)$;

e3= $Bus1_{x_l\ IN\ Station3-Station4}(x_v > 7)$;

e4= $Bus1_{x_l\ IN\ Station4-Station5}(x_v > 6)$.

3.2. The Processing Algorithm of Monitoring Event

We use the event identify trees algorithm to detect the complex event. And the main idea of the detection algorithm is that: we express the complex event expression as an event tree. The base events are the leaf nodes of the event tree and non-leaf nodes represent the complex events. If an event instance of the event stream meets the condition of the leaf node of the event tree, we mark this node of true. If all sub-nodes of a non-leaf have been marked true, we should judge the condition of the non-leaf node and marked it by true or false. If the root-node of the event tree is true, we can consider that the complex event has been successful detected.

Based on the above ideas, the complex event detecting algorithm as following:

Table 2. The Complex Event Detecting Algorithm Based Event Tree

```
input：complex event tree and the event instances flow
output：the information of the complex event was successful matched

Message ComplexEventMatchTree(MatchTree tree, EventFlow EventInstance)
{
      for(every EventInstance)
      {
            if(EventInstance.ID=Tree.LeafNode.ID)
               {
                    if(GetMatched(EventInstance,Tree.LeafNode)=True)
                    Tree.LeafNode=True;
                  activeLeafNode(Tree.LeafNode);
                }
       }
}
activeLeafNode( LeafNode )
{
      for(every LeafNode has MidNode)
      {
            if(GetMatched(Tree.LeafNode,Tree.MidNode)=True);
               {
                 Tree.MidNode=True;
                activeMidNode(parentNode, MidNode);
               }
      }
}
activeMidNode(parentNode, subEvent)
{
  for(every parentNode)
  {
      if(parentNode !=root)
        {
           if(GetMatched(subEvent, parentNode)=True)
             {
              parentNode=True;
              activeMidNode(parentNode.getParent( ), parentNode.genCEvent( ));
             }
             continue;
      }
    else
```

```
        {
                Tree.RootNode=True;
                Return("Match Succeed!");
            }
    }
}
```

Time complexity analysis: In the above algorithm, assume that there are n nodes in the complex event tree. If detect a complex event expression, we should traverse the event tree. So the time complexity of the algorithm is o(n).

4. The Vehicle Monitoring System

4.1. The Architecture of the Vehicle Monitoring System

In internet of vehicles, the vehicles have the ability to compute and communicate. There are the roadside units (RSU) lay on both sides of the road. And the RSUs also have the ability to compute and communicate. The vehicle monitoring centre completes the vehicle monitor via the communication with the vehicles and the RSUs. The architecture of the monitoring system as shown below:

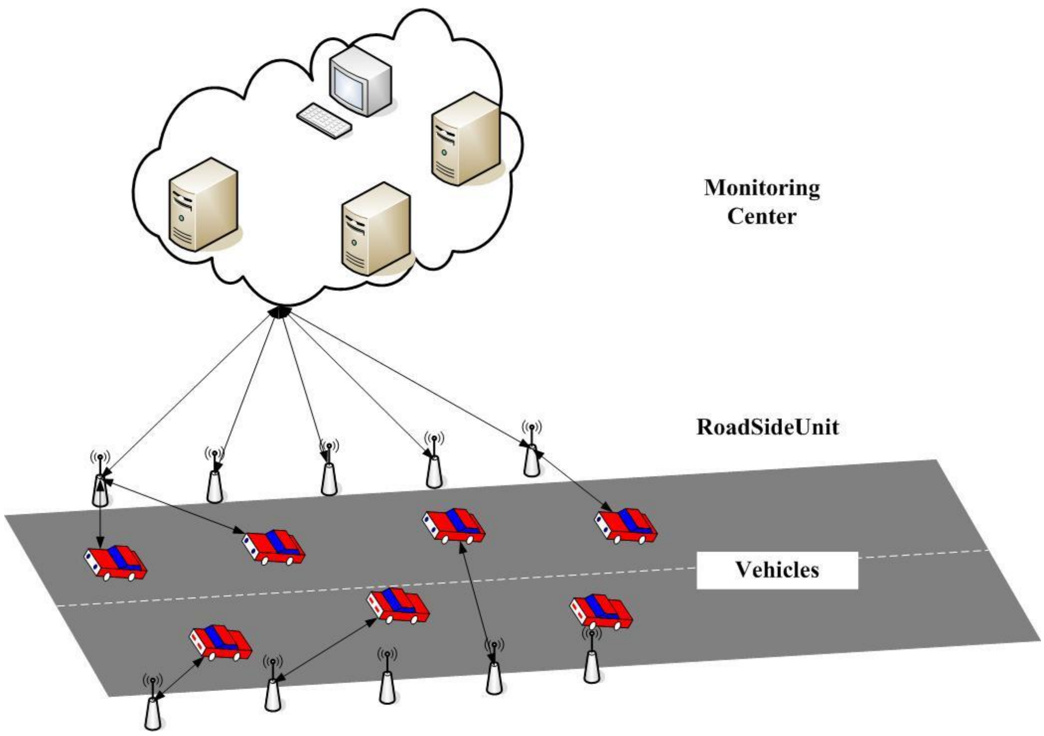

Figure 2. The Architecture of The Vehicle Monitoring System

As seen from the Figure 2, the vehicle monitoring system is composed of three parts: vehicles, roadside units and the vehicle monitoring center. The vehicles and the roadside units contain embedded devices, which have the computing capacity. There are many large servers, routers and other equipments in the vehicle monitoring center and these equipments have strong computing power. The vehicles can communicate with each other. And they can also communicate with the roadside units. Their major communication methods are wireless. The roadside units can also communicate with the vehicle monitoring center by the wired connection. Of course, the vehicles can directly communicate with the monitoring center by the 3G or 4G wireless networks. But this is not the main method of their communications.

4.2. The Monitoring Strategy of the Vehicle Monitoring System

Based on the above vehicles monitoring system of internet of vehicles, we present an event-based vehicle monitoring strategy:

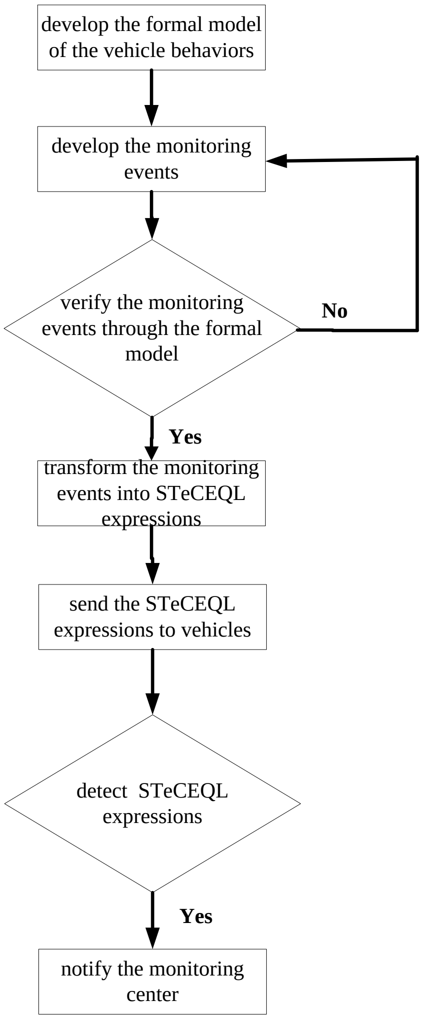

Figure 3. The Vehicle Monitoring Strategy

The traditional vehicle monitoring system needs transmitted all the data of the running vehicles to the monitoring center. Afterward the monitoring centers' computers will process these data. This traditional vehicle monitoring strategy gives the great pressure on the data transmission. And with the increasing of the vehicles, the monitoring center computing capacity cannot meet the actual demand. The event-based vehicle monitoring system does not have to put all the vehicles' sensor data back to the monitoring center database. The vehicle monitoring center will sent the complex event query expressions to the embedded computers of the vehicles. Then the vehicles can real-time processing its data by the complex event expressions. Finally, the data of vehicles will be transformed into the event

5. The Simulation and Analysis of the Results

In order to verify the effectiveness of the vehicles monitoring strategy, we have design an experiment based on the bus system of the Figure 1 in the Section 2. The grid map of the experiment is shown in Figure 1, we have simulated two different buses data. These bus data is shown as below:

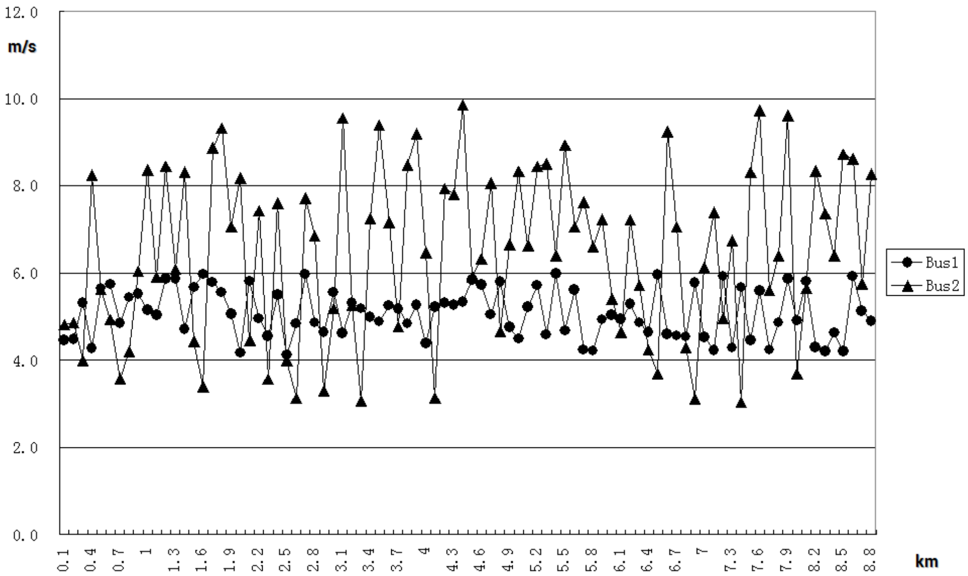

Figure 4. The Simulated Bus Data of The Experiment

In Figure 4, the abscissa represents traveling distance of the bus, and the vertical axis indicates the traveling speed of the bus. The data point of the figure represents the bus speed have changed by the traveling distance. As the figure shows, Bus1 is running smoothly than Bus2.

We have developed the monitoring events: the speed of the bus could not more than 5m/s.According to the previous monitoring strategies in Figure 3, we finally get the results of the monitoring events of Bus1 and Bus2 as in Figure 5 and Figure 6.

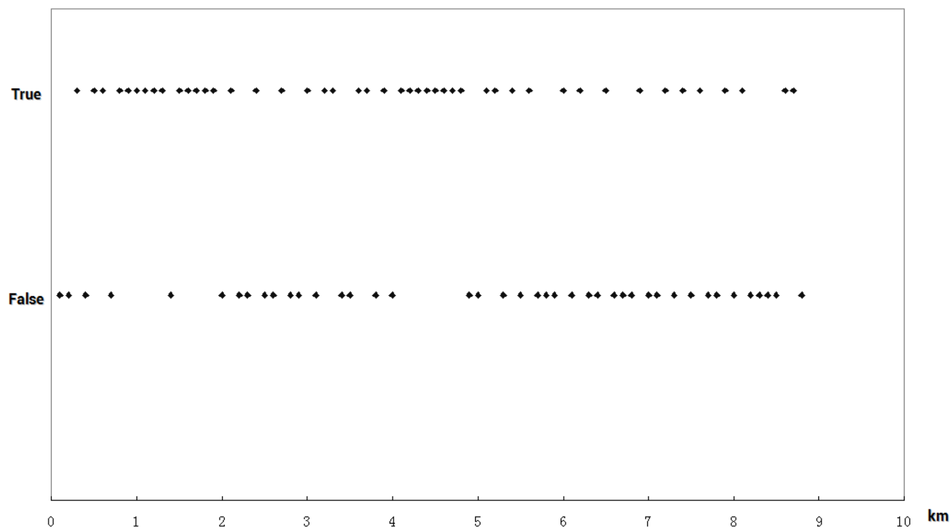

Figure 5. The Monitoring Results of The Bus1

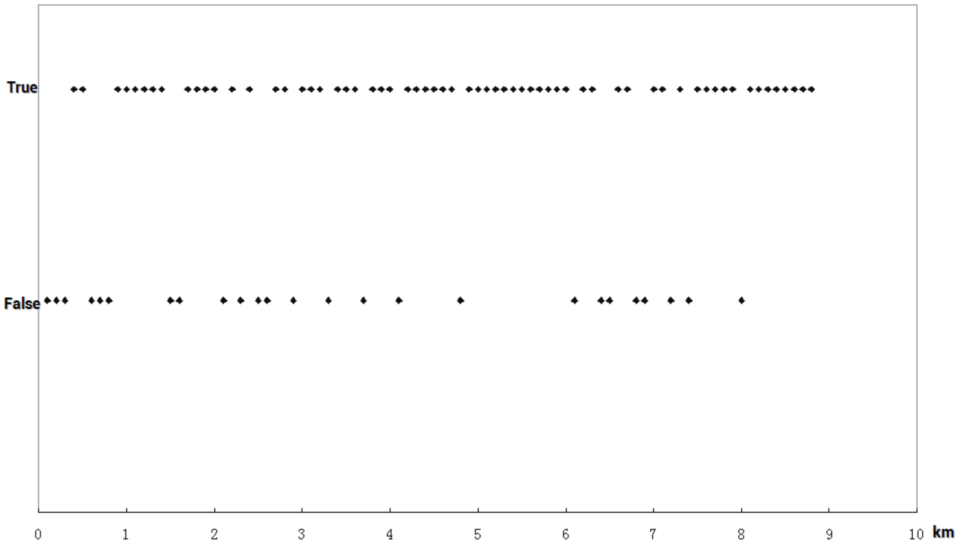

Figure 6. The Monitoring Result of Bus2

In Figures 5 and 6, the horizontal axis represents the distance of the buses and the vertical axis has two values: True and False. The true indicates the monitoring event has occurred and the false indicates the monitoring event has not occurred. From the experimental results, we can see the effectiveness of the vehicle monitoring strategy.

6. Conclusions

In this paper, we propose a vehicle monitoring system based event and the corresponding monitoring strategy. In this type vehicle monitoring system, the monitoring event will be developed by the monitoring center. Then the monitoring event will be converted to the corresponding complex event expression and transmitted to the embedded system of the vehicles. In the embedded system of the vehicles, the vehicles' data will be real-time process by the STeCEQL. Finally, we demonstrate the effectiveness of the vehicle monitoring strategy through the simulation.

.

Acknowledgements

This work is supported by the National Basic Research Program of China (No. 2011CB302802), the National Natural Science Foundation of China (No.61370100 and No.61021004) and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

[1] G. Dimitrakopoulos, "Intelligent transportation systems based on internet-connected vehicles: Fundamental research areas and challenges", IEEE, **(2011)**.
[2] Y. Chen, "Stec: A location-triggered specification language for real-time systems", IEEE, **(2012)**.
[3] H. Wu, Y. Chen and M. Zhang, "On Denotational Semantics of Spatial-Temporal Consistency Language—SteC", IEEE, **(2013)**.
[4] J.H. Hu, W. Tang and H.Y. Xie, Advanced Materials Research, vol. 433 **(2012)**.
[5] X. Wang and M. Wang, "Integrating GIS, GPS technologies for designing vehicle monitor system", IEEE, **(2010)**.
[6] T. Chakravarty, K. Srinivasarengan, S. Roy, S. Bilal and P. Balamuralidhar, "A model-based approach to monitor complex road-vehicle interactions through first principles", IOP Publishing, **(2013)**.
[7] A. Aljaafreh, M. Khalel, I. Al-Fraheed, K. Almarahleh, R. Al-Shwaabkeh, S. Al-Etawi and W. Shaqareen, "Vehicular data acquisition system for fleet management automation", IEEE, **(2011)**.
[8] S. Schwiderski-Grosche and K. Moody, "The SpaTeC composite event language for spatio-temporal reasoning in mobile systems", ACM, **(2009)**.

[9] K. Moody, J. Bacon, D. Evans, S. and Schwiderski-Grosche, "From active data management to event-based systems and more", Springer, **(2010)**.
[10] B. Jin, W. Zhuo, J. Hu, H. Chen and Y. Yang, "Decision Support Systems", vol. 55, **(2013)** .

Authors

Huiyong Li received his M.Sc. in Computer Sciences (2011) from Taiyuan University of Science and Technology. Now he is PhD student of software engineering institute, East China Normal University. His current research interests include different aspects of Internet of Things and Real-time Distributed Systems.

Yixiang Chen received his PhD in Mathematics (1995) from Sichuan University. Now he is Chair of the MoE Key Research Center for Software/ Hardware Co-design Engineering, East China Normal University. His current research interests include different aspects of Real-time System, Formal Models, Formal Semantics of Programming, Foundation of Computations and Trustworthy Network.