ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Online Toll Tax Database Management Model through Push-Down Automata

Ravi Pratap Singh[1], Dr. Vivek Kr. Singh[2]

M. Tech, Dept. of Computer Science and Engineering (Software Engineering), BBDU Lucknow, U.P., India[1]

Professor, Dept. of Computer Science and Engineering, School of Engineering, BBDU, Lucknow, U.P., India[2]

ABSTRACT: This paper highlight the use of Push-Down Automata in maintaining the vehicle records and provide the clustered view of them to make comparative analysis easier and faster. We focus to provide a communicative framework that can record the vehicles coming from a particular state. The Object Constraint Language (OCL) is being applied on Object Oriented Language so that the communication framework can be represented into Object Oriented Language. In order to achieve this, the Real Time Constraint Notation being applied to the push-down automation for formal verification of the model.

KEYWORDS*:* Two-Stack Push Down Automata (PDA), Object Constraint Language (OCL), Real Time Constraint Notation (RTCN).

I. INTRODUCTION

Tolling as a method of financing the transportation system is becoming more common day by day. Neither the traveling public nor State Departments of Transportation want vehicles to stop or slow down to pay to use a toll facility. In this project we highlight the use of Push-Down Automata (PDA) in sorting and maintaining the vehicle records coming for which state, calculate the total revenue generated and showing this information into cluster view of vehicle logs. Our objective is to develop a more interactive and communicative framework that can maintain a record of the vehicles coming from a particular state. Since Object Constraint Language is being applied on object-oriented language we will be representing the entire communication framework into Object-Oriented Language.

In order to achieve this goal we have identified these basic steps, the first step deals with the development of ubiquitous computing environment as the vehicle coming from. Secondly, development of push down automata model for vehicle logs database management system. The proposed work of this research work can be described as following points:

- Highlight the use of PDA in sorting and maintaining record.
- Focus is to provide a communicative framework that can record vehicles log.
- RTCN being applied to the PDA for formal verification of the proposed model.

For this, there will be three basic steps:

- Development of Ubiquitous computing environment.
- Development of PDA model for database management system.
- To verify the model and do the performance analysis.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

II. RELATED WORK

I went through various research papers for literature review, some of those and their related work are illustrated as below:

- **Enhancement of User's Call Logging facilities using Push Down Automata (PDA) with Real Time Constraint Notation (RTCN):**

This paper highlights the use of Push Down Automata in maintaining the call logs. The special feature about this paper is to maintain a log of incoming calls from the different mobile service provider and represent in the clustered way for user. This paper focuses on the development of a feature in mobile phone for user to view number of incoming calls in cluster view to know how many calls are received from different mobile service providers. This process can also be implemented for outgoing calls & helps to mobile service providers to analysis comparative calls rate from different mobile service provider. [1]

- **Putting 00 to Work: Results from Applying the Object-Oriented Paradigm during the Development of Real-Time Applications:**

This position paper gives an overview of some results and lessons we have learned from the application of object-oriented techniques to real-time system development. These results have been investigated about what advantages can be achieved applying 00 issues to real-time system development and what improvements in the existing 00 method are required to make them really usable for real-time applications. The major lesson learned from this is that existing object-oriented approaches are still not solution for the development of real-time applications, mainly because a mechanism to deal with timing semantics is still lacking. The results of the comparison between the real-time software obtained using a high-level real-time programming language and an object-oriented one confirmed that assumption. Certainly, by adding these real-time features to object oriented concepts a very powerful mechanism to describe and implement real-time applications can be achieved. [14]

- **Computer Vision Based Vehicle Detection for Toll Collection System Using Embedded Linux:**

This paper presents a brief review of toll collection systems present in India, their advantages and disadvantages and also aims to develop a new efficient toll collection system that would be a good low cost alternative among all other systems. This proposed system is based on Computer Vision vehicle detection using OpenCV library. The system is designed using Embedded Linux development tool kit. Embedded Linux platform is very useful for implementing different aspects of the proposed system. Ubuntu provides a powerful interface between Open CV and lower level peripherals like GPIOs, therefore embedded hardware can be controlled from image processing programs written using Open CV. [2]

- **Toll Snapping and Processing System:**

In this paper automation in toll tax payment using Online Registration form and RFID is proposed. Automation of toll plaza is made using the combination of microcontroller, RFID. The proposed RFID system uses tags that are mounted on the windshields of vehicles, by means of which information embedded on the tags will be read by RFID readers. The toll authorities and motorists can easily exchange the important data information, there by enable a more efficient toll collection by reducing traffic and eliminating possible human errors. [3]

- **Selection of optimal electronic toll collection system for India: A subjective-fuzzy decision making approach:**

Present study deals with the adoption of newer technologies for developing countries. Most of the developing nations due to lack of resources perform techno-socio-economic analysis on the already existing models of the developed ones. Such adopted techniques might not be performed effectively because of unlike socio-economic factors. So, it becomes crucial to select new technologies based on appropriate and more suitable criteria with respect to a particular nation. In this paper, the selection of optimal electronic toll collection (ETC) system for India is demonstrated. For this context, thirteen crucial parameters for selection of appropriate ETC system are considered. This present study gives the feasibility of fuzzy VIKOR method in policy formulation and adoption of optimal technologies as per infrastructure and economic conditions of a nation. [9]

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

- **Raspberry Pi Image Processing Based Economical Automated Toll System:**

The highway toll system has already been developed and widely being used in many developed nations, but most of them uses Radio Frequency ID, but in developing countries RFID for each vehicle doesn't exist and using RFID is still a costlier solution. The image processing technique to detect license plate for auto toll system are being used by some of the developing countries but the problem is not solved yet because of high cost of host device (e.g. computer) to run. Keeping those problems into mind, in this paper a new approach is proposed in which raspberry pi will be used as the host. This minicomputer has ability of image processing and controlling a complete toll system. [17]

- **Image Processing Based Automatic Toll Booth in Indian Conditions:**

In this research paper the image and respective information is processed based on toll collection system and focuses on how to make more efficient and perfect. The proposed system would help to pay the toll automatically and reduces the queue at the toll booth. In this system camera is used for capturing the image of the vehicle number plate and then the captured image would be converted into the text using ANPR (Automatic Number Plate Recognition) technique and the toll would be deducted from the customer's account and then open the gate. [10]

III. AN INTRODUCTION TO TWO-STACK PUSH DOWN AUTOMATA

A pushdown automata (PDA) is a type of automaton that employees a stack. They are less capable than turing machines but more capable than finite-state machines. A pushdown automaton may be pictured as a finite automaton with the stack in which the symbols may be 'pushed' or may be 'popped'. A normal PDA has one stack for controlling the parsing of the input string and the input tape will move left one cell on each input. It is highly impossible to support and solve all languages of context free grammar with normal PDA. Such as, the context free languages like $L = \{a^n b^n c^n; n \geq 1\}$ the normal PDA with one single stack will not work and hence we need another option, which might be more complex and time taking than PDA. These problems can be solved by increasing the number of stacks in PDA. The number of stacks may vary according to the requirements of input given (input alphabets) and the parser machine. As per requirement, if needed to solve complex problems of CFL the number of stacks in PDA could be increased. The Time Complexity of the stack operations (i.e. push and pop) is O(1); therefore by increasing the number of stacks on a PDA won't affect more in the efficiency. The push and pop operation can be performed in a single iteration and hence the complexity of the PDA will not high.

A. *Formal Definition of Two-stack PDA:*

Two-stack PDA is a stack variation of normal PDA, the number of stacks present in the automata machine PDA is the only difference. The two stacks PDA machine can be represented as:

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, Z_1, F)$ where

- Q is a finite set of states
- Σ is a finite set of the input alphabets
- Γ is a finite set which is called the stack alphabet
- δ is a finite subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$, the transition relation.
- $q_0 \in Q$ is the start state
- $Z_0 \in \Gamma$ is the initial stack symbol of stack1
- $Z_1 \in \Gamma$ is the initial stack symbol of stack2
- $F \subseteq Q$ is the set of final state/states

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

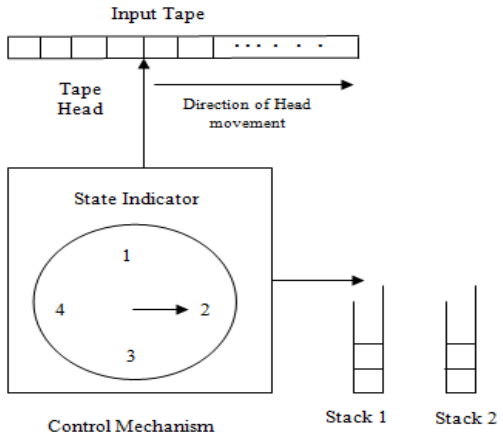

Fig. 01: Two-Stack Push Down Automata Model

B. *Transitions in PDA:*

A transition in PDA is depends on:

- The current state of the machine
- The symbol read from the input tape
- The top symbol on the stack

Outcomes of the transitions are:

- After reading the current state of input alphabet it may be transit to new state or on same state (loop condition).
- The new stack alphabet may be pushed or current top of the stack alphabet may be popped on the top of the stack.
- No change on the top of the stack.

C. *Instantaneous Description(ID):*

At a given instant of time the configuration of PDA is called instantaneous description (ID). It is defined to be a member of Q x $\sum$ x Γ^* where The first component is the state of the machine, the second is an input yet to be read and third is the contents of the stack i.e. $\forall$ (q, z) $\in \delta$ (p, a, Z) and $\forall$ w $\in \sum^*$ and $\alpha \in \Gamma^*$ we define:

(p, aw, Zα) $\vdash$ (q, w, za)

IV. AN INTRODUCTION TO REAL TIME CONSTRAINT NOTATION (RTCN)

A Real-time constraint notation are used to provide sufficient real-time specifications in object-oriented language as one may expect from a real-time language, while in the object-oriented tapestry integrating these specifications. The Real Time Constraint Notation (RTCN) is used to represent and model the constraints in real time modeling. The significant aspects in object oriented real-time modeling can be identified as:

- The use of inheritance and redefinition of real time constraint through the inheritance hierarchy and extension of the inheritance of the state and behavior of a class to include the definition of temporal constraints.
- The reuse of the temporal constraints specifications through the inheritance mechanism and across class boundaries.
- Time-abstraction is a natural approach for specifying timing constraints, where the class is defined with the constraint definition and then class implementation associated with it.

Some of the major characteristics of RTCN are:

- Sequence Diagram and Message Sequence Charts.
- Scenario Composition
- Defining Finite State Machine and State Charts

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

- Explaining State Chart Synthesis
- Representing Sequence Diagram Composition
- Deterministic Grammar

V. AN INTRODUCTION TO OBJECT CONSTRAINT LANGUAGE (OCL)

The Object Constraint Language is a modeling language with which we can build software models. OCL is a standard "add-on" to the Unified Modeling Language. In OCL, every written expression will rely on the type (i.e. the classes, interfaces and so on) and those are defined in the UML diagrams. In the model to indicate a value the OCL expressions can be used anywhere. It is based on mathematical set theory and predicate logic. OCL has a formal mathematical semantics. The development of software models are utilizing the OCL, hence regarded as a modeling language. OCL when used along with UML supports a lot more additional features. OMG is a standard for analyzing and designing in object oriented manner. Each expression which is represented in OCL is based on the types (whether classes or interfaces or any other) which are explained in the UML diagrams. Therefore UML will definitely be used in the OCL based applications. A strong relation exists among the OMG standards. Models serve as building block for the development of software, a significant feature of MDA. Thus it is necessary that we have consistent models. A combined effort of OCL and UML produces consistent models.

The application of constraints in the UML diagrams is the major focus of OCL. Object Oriented Languages and their semantics are well modeled by using Object Constraint Language (OCL). For the modeling of real time systems the OCL integrated UML is being utilized. Real time systems properties are well represented by application Computational Tree Logic (CTL). These types of approaches are highly applicable for the formal verification of the develop models. OCL has been utilized in various forms:

- Specification of constraint on operations.
- To define guard.
- As a navigation language.
- To define pre and post- conditions on Operations and Methods.
- Specification of type invariants for Stereotypes.
- Specification of invariants on classes and types in the class model.

Characteristics of OCL: A model has more than one class, required OCL statements to be a consistent model. With the help of UML diagrams many inconsistencies would persist. Therefore, OCL is a necessary language for developing efficient models.

- Query and Constraint Language Description
- Foundation with Mathematical Logic
- Strongly Typed Language
- Declarative Language

VI. PROPOSED WORK

A. *Flow Chart of Proposed PDA Model:*

Working of two-stack PDA model for maintaining record of the vehicles passing through a Toll Plaza in clustered way can be represented by the flow chart as shown in Fig. 02 :

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

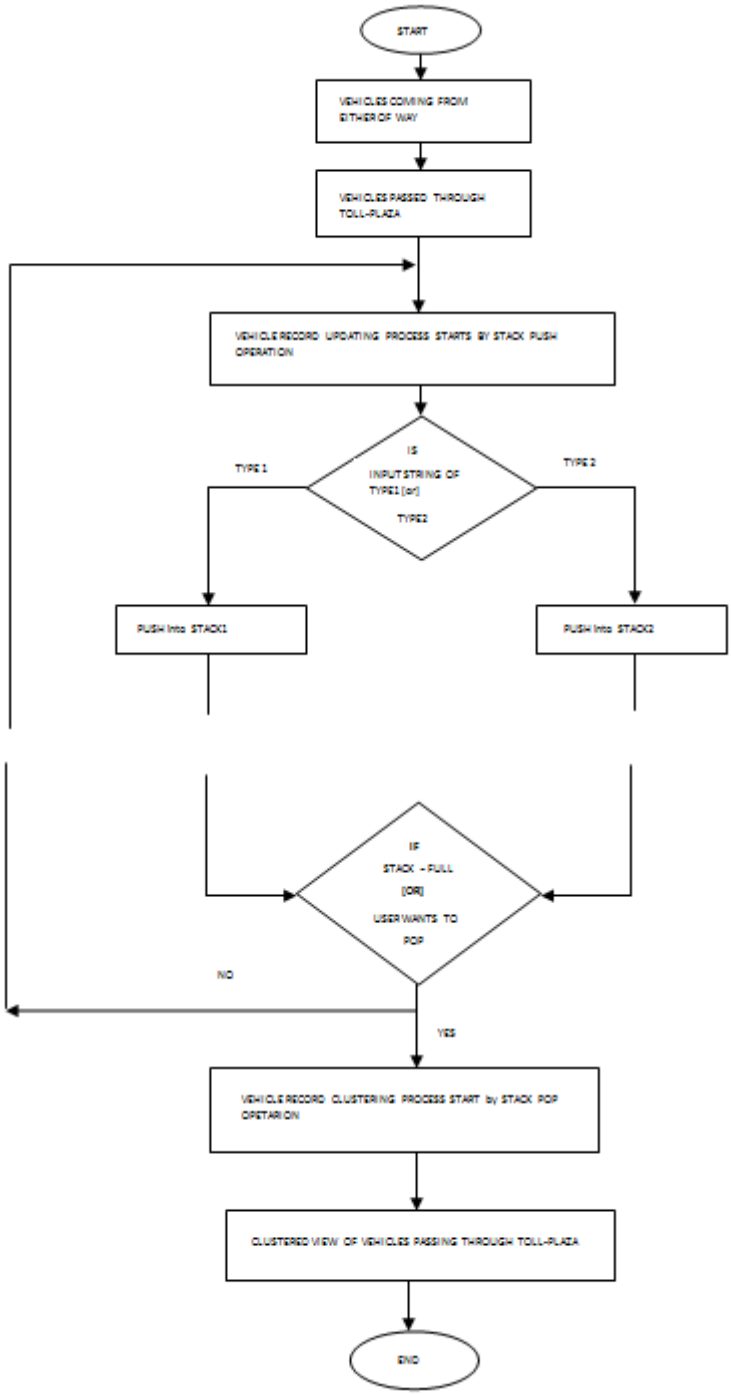

Fig. 02: Flow Chart of working of Two-Stack PDA model for vehicle record passing through Toll-Plaza

B. *Proposed Algorithm:*

```
While (input != empty)
{
    Read input[i]
```

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

```
        If (input[i] = type1)
        {
                If (stack2 != empty)
                        Change state
                Push (input[i]) in stack1
                Count1 ++
        }
        Elseif (input[i] = type2)
        {
                If (is first iteration)
                        Change state
                Push (input[i]) in stack2
                Count2 ++
        }
        Else
        {
                If (is first iteration)
                        Change state
                If (stack1 = empty || stack2 = empty)
                        Break
                Else
                {
                        Pop (Stack1)
                        Pop (Stack2)
                }
        }
   }
If (input tape = empty && stack1 = empty &&stack2 = empty)
        Input string accepted
Elseif (input tape = empty && count1 = count2)
{
                Free (stack1)
                Free (stack2)
                Input string accepted
        }
        Else
                Input string is rejected
        STOP.
```

C. *Steps Involved:*

The steps which are mentioned below can be modeled with the help of two-stack PDA. The steps are following:

1. Read input tape
2. Check input type
   - 2.1. If input type is of type1 then push(stack1)
   - 2.2. Change state if the stack 2 in NOT empty
   - 2.3. Increment count1 by one
3. if input type = type2
   - 3.1. Change the state if it is first iteration of type2
   - 3.2. Push input value into the stack2
   - 3.3. Increment count2 by one

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

4. Elseif change state
   4.1. If this is the first iteration of other type
   4.2. Check stacks if the stacks are empty then operati
   4.3. Else if POP(stack1) and POP(stack2)
5. Follow above steps till the input tape = Empty
6. If the input tape is ended and both stacks are empty then accept input
7. If the input tape is empty and count1 = count2 then accept input
8. If the above operations fail then reject it.

D. *PDA Transitions for Vehicle Record Update:*

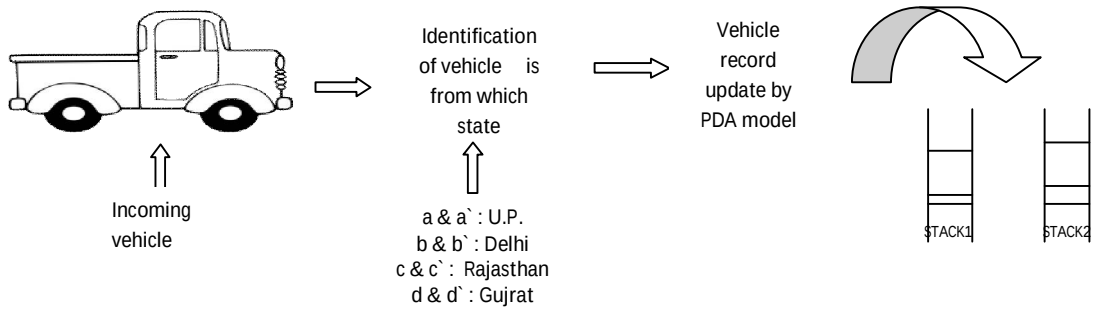

Fig. 03: Incoming vehicles record update process

Let us assume the input string is abcda`b`c`d`. Now, to achieve this goal, we design the two-stack PDA as:
$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, Z_1, \phi)$ where

- Q: is the set of states for maintaining the vehicles record passing through toll plaza i.e.
  $Q = \{q_0, q_{PUSH}, q_{POP}\}$
- Σ: is the finite set of input alphabets for vehicles from different states i.e.
  $\Sigma = \{a, b, c, d, a`, b`, c`, d`\}$ where:
  - a and a` represents: vehicles from U.P.
  - b and b` represents: vehicles from Delhi
  - c and c` represents: vehicles from Rajasthan
  - d and d` represents: vehicles from Gujrat
- Γ: is finite set of stack alphabets for vehicles from different states i.e.
  $\Gamma = \{Z_0, Z_1, Z_a, Z_b, Z_c, Z_d\}$ where:
  - Z_0: is the initial stack alphabet of stack1
  - Z_1: is the initial stack alphabet of stack2
  - Z_a: is the stack alphabet PUSHED into or POPPED from stacks when vehicle is from U.P.
  - Z_b: is the stack alphabet PUSHED into or POPPED from stacks when vehicle is from Delhi
  - Z_c: is the stack alphabet PUSHED into or POPPED from stacks when vehicle is from Rajasthan
  - Z_d: is the stack alphabet PUSHED into or POPPED from stacks when vehicle is from Gujrat

For input string as mention above the PDA transitions are:

i. $\delta(q_0, a, Z_0, Z_1) = (q_{PUSH}, Z_aZ_0, Z_1)$
ii. $\delta(q_{PUSH}, b, Z_a, Z_1) = (q_{PUSH}, Z_bZ_a, Z_1)$
iii. $\delta(q_{PUSH}, c, Z_b, Z_1) = (q_{PUSH}, Z_cZ_b, Z_1)$
iv. $\delta(q_{PUSH}, d, Z_c, Z_1) = (q_{PUSH}, Z_dZ_c, Z_1)$
v. $\delta(q_{PUSH}, a`, Z_d, Z_1) = (q_{PUSH}, Z_d, Z_aZ_1)$
vi. $\delta(q_{PUSH}, b`, Z_d, Z_a) = (q_{PUSH}, Z_d, Z_bZ_a)$
vii. $\delta(q_{PUSH}, c`, Z_d, Z_b) = (q_{PUSH}, Z_d, Z_cZ_b)$
viii. $\delta(q_{PUSH}, d`, Z_d, Z_c) = (q_{PUSH}, Z_d, Z_dZ_c)$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

E. *Instantaneous Description (ID) for Vehicle Record Update:*

As per the rules, the ID for input string as mention above is given below:

q_0, abcda`b`c`d`, Z_0, $Z_1 \vdash q_{PUSH}$, bcda`b`c`d`, a$Z_0Z_1 \vdash q_{PUSH}$, cda`b`c`d`, baZ_0, $Z_1 \vdash q_{PUSH}$, da`b`c`d`, cbaZ_0, $Z_1 \vdash q_{PUSH}$, a`b`c`d`, dcbaZ_0, $Z_1 \vdash q_{PUSH}$, b`c`d`, dcba$Z_0$, a`$Z_1 \vdash q_{PUSH}$, c`d`, dcbaZ_0, b`a`$Z_1 \vdash q_{PUSH}$, d`, dcba$Z_0$, c`b`a`$Z_1 \vdash q_{PUSH}$, ε , dcbaZ_0, d`c`b`a`Z_1

F. *PDA Transitions to generate Cluster View of Vehicle Record:*

The vehicle log record update process, as shown in Fig. 03, will continue till stack is full or user itself wants to pop. When either of the two pop conditions occurred, then transitions for cluster view generation start according to the following steps:

Step 1: When the pop condition came then $ symbol is generated on the input tape and q_{PUSH} is converted to q_{POP}

i. $\delta(q_{PUSH}, \$, d, d`) = (q_{POP}, d, d`)$
ii. $\delta(q_{PUSH}, \$, c, c`) = (q_{POP}, c, c`)$
iii. $\delta(q_{PUSH}, \$, b, b`) = (q_{POP}, b, b`)$
iv. $\delta(q_{PUSH}, \$, a, a`) = (q_{POP}, a, a`)$

Step 2: The following transitions occurred for the generation of cluster view with the help of STACK POP operations as follows:

i. $\delta(q_{POP}, \$, a, a`) = (q_{POP}, \varepsilon, \varepsilon)$
ii. $\delta(q_{POP}, \$, b, b`) = (q_{POP}, \varepsilon, \varepsilon)$
iii. $\delta(q_{POP}, \$, c, c`) = (q_{POP}, \varepsilon, \varepsilon)$
iv. $\delta(q_{POP}, \$, d, d`) = (q_{POP}, \varepsilon, \varepsilon)$

Step 3: When the top of stack1 = Z_0 and stack2 = Z_1, it implies that all the stack alphabets are popped out corresponding to different states and Z_0 and Z_1 are the top of stack1 and stack2 respectively then the state q_{POP} is transit to q_{PUSH} and the PDA model is ready to update the new vehicle log record into stacks. Step III condition in two-stack PDA model could be completed with the help of following transition:

i. $\delta(q_{POP}, \$, Z_0, Z_1) = (q_{PUSH}, Z_0, Z_1)$

VII. VERIFICATION USING OBJECT CONSTRAINT LANGUAGE (OCL)

A constraint is defined as “restriction on one or more values of an object – oriented model or system”. In order to apply constraints on database, we have OCL invariants and using this on database table consisting of details of the vehicles coming and their number. For that, transformation pattern is given as:

Context: Session
Inv: for coming vehicles
Inv: Self.Stack1 ↛ Empty( )
Inv: Self.Stack2 ↛ Empty( )
Pre: Self.Stack1 = Q
Pre: Self.Stack2 = Q
Post: Self.Value = Γ

Context: Record Call
Session.Record =
(if q_{PUSH} != a or a` then ‘Z_a’ or
b or b` then ‘Z_b’ or
c or c` then ‘Z_c’ else
d or d` then ‘Z_d’)

Context: Cluster View Generation
Session.Cluster View =
(if STACK = FULL then $q_{PUSH} = q_{POP}$
end if)
Session.READ =

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

(if Input string = abca`db`ac` then
Count := 03 for U.P. and
Count := 02 for Delhi and
Count := 01 for Rajasthan and
Count := 01 for Gujrat
end if)

VIII. BENEFITS AND RESULT OF PROPOSED MODEL

In this project we develop an interactive and communicative framework that can maintain a record of the vehicles coming from various states. It focuses on development of a feature to view vehicles coming from which territory in cluster view. We can calculate the total revenue generated territory wise and generate a cluster view of vehicle logs. It helps in comparative analysis, having the following importance:

- Financial leakage control
- Vehicle tracking
- Congestion Management
- Low cost and easy to implement

Human Visualization: The working of Push-down Automata model for vehicle passing through the particular highway records storage in cluster way and can be represented by the graphical way as well.
Better Interaction and Analysis: The interaction between the user and the framework will be helpful to achieve better results and this interaction ranges from allowing user simply to calculate the total tax generated from a specific state.
Efficient Result: In general, the automata theory always provides the more efficient result comparatively. By model checking we will analysis and prove that the developed framework giving result more efficiently in comparison to other existing models.

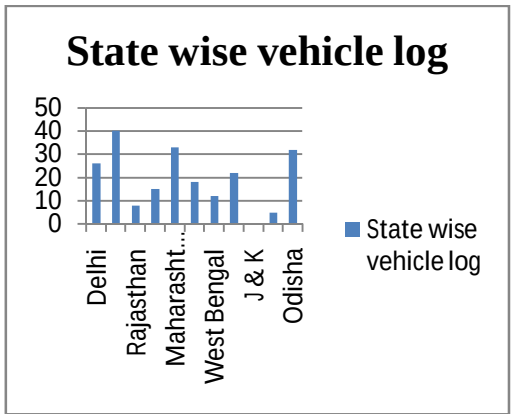

Fig.04: Resulted Clustered View of proposed model

IX. CONCLUSION

This research work focuses on development of a feature to view vehicles coming from which territory in cluster view and show this into cluster view of vehicle logs. This clustered view will help in comparative analysis. It is low in cost and easy to implement that will support into financial leakage control. In future, this kind of interactive framework can also be maintained for Airlines Database Management model for domestic and international flights as well. Further, this model can be enhanced on the Ship Database Management for maintaining the log of different ships for the purpose of comparative analysis.

Acknowledgment— I would like to express my keen thanks and heartily gratitude to Dr. Vivek Kr. Singh**,** Department of Computer Science and Engineering, School of Engineering, BBDU, Lucknow, for offering much appreciated advice, continuous support and thought provoking ideas throughout in carrying out the project. It is solely

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

his motivation that has driven me in efforts. I would like to express my sincere thanks to all my colleagues whose periodic inputs and informal discussions have given me a better perspective of my progress.

REFERENCES

1. Vivek Kr. Singh, S. P. Tripathi, J. B. Singh and R. P. Agarwal, 'Enhancement of User's Call Logging facilities using Push Down Automata with Real Time Constraint Notation', International Journal of Computer Science Issues(IJCSI), Vol. 9, Issue 3, pp. 216-220, 2012.
2. Mr.Abhijeet Suryatali and Mr. V. B. Dharmadhikari, 'Computer VisionBased Vehicle Detection for Toll Collection System Using Embedded Linux', International Conference on Circuit, Power and Computing Technologies [ICCPCT], IEEE, pp. 978-984, 2015.
3. Padmavathi Shyadambi, Saikumari S, Jyoti Chitti, Deepika Paragoudar, Raghavendra Havin and Praveenkumar Hadapad, 'Toll Snapping And Processing System', International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE), Vol. 14, Issue 2, pp. 63-66, 2015.
4. Abel Shajan, Abin Jose and Kalung Tari, 'Automatic Toll Collection System', International Journal of Emerging Trends in Science and Technology (IJETST), Vol. 02, Issue 04, pp. 2242-2247, 2015.
5. Devadasu Saraswathi And P. Suresh Kumar, 'Electronic Toll CollectionSystem and GSM Trace for Smart Vehicles', International Journal of Scientific Engineering and Technology Research, Vol. 04, Issue 28, pp.5419-5424, 2015.
6. P. Siva Kumar and R. S. Pratap Singh, 'Implementing ALPR System using Vehicle LP Recognition Information', International Journal of Scientific Engineering and Technology Research, Vol. 04, Issue 15, pp. 2744-2746, 2015.
7. Marco Amorim, Antonio Lobo, Carlos Rodrigues and Antonio Couto, 'Optimal location of electronic toll gantries: The case of a Portuguese freeway', ScienceDirect, pp. 880-889, 2014.
8. Nale Zhao, Tongyan Qi, Lei Yu, Juwen Zhang and Pengpeng Jiang, 'A Practical Method for Estimating Traffic Flow Characteristic Parameters of Tolled Expressway Using Toll Data', ScienceDirect, The 9th International Conference on Traffic & Transportation Studies (ICTTS'2014), pp. 632-640, 2014.
9. Saurabh Vats, Gaurav Vats, Rahul Vaish and Varun Kumar, 'Selection of optimal electronic toll collection system for India: A subjective-fuzzy decision making approach', Elsevier, Applied Soft Computing 21, pp. 444-452, 2014.
10. Priyanka Chhoriya, Govinda Paliwal and Poonam Badhan, 'Image Processing Based Automatic Toll Booth in Indian Conditions', International Journal of Emerging Technology and Advanced Engineering, Vol. 3, Issue 4, pp. 2250-2459, 2013.
11. Navin Kumar Chaudhary, Rabin Karmacharya, Binaya Ghimire, Dr.N. Srinivasu, 'Stack Variation In Push Down Automata', International Journal of Engineering Trends and Technology (IJETT), Vol. 04, Issue 5, pp. 1535-1539, 2013.
12. Bihari T., Gopinath P. and Schwan K., 'Object-Oriented Design of Real Time Software', IEEE Software Trans., pp. 194-201, 1989.
13. Dr. S. P. Tripathi, Prof. J. B. Singh, Vivek Kr. Singh, 'Design Mobile Data Logging Framework using Ubiquitous Computing Environment', International Journal of Hybrid Information Technology, Vol. 2, No. 4, pp. 19-24, 2009.
14. Pereira C., 'Putting OO work: Results from Applying the Object-Oriented Paradigm during the Development of Real Time Applications', Fifth Euromicro Workshop on Real-time Systems Proceedings, pp. 166-170, 1993.
15. Md. Farhad Ismail and M.A.R. Sarkar, 'Development of a Model for Electronic Toll-Collection System', I.J. Intelligent Systems and Applications, Vol. 01, pp. 39-45, 2012.
16. Kligerman E. and Stoyenko A., 'Real Time Euclid: a Language for Reliable Real Time Systems', IEEE Trans. on Software Engineering, Vol. SE-12, No. 9, pp. 941-949, 1986.
17. Md. Kawser Jahan Raihan, Mohammad Saifur Rahaman, Mohammad Kaium Sarkar and Sekh Mahfuz, 'Raspberry Pi Image Processing Based Economical Automated Toll System', Global Journal of Researches in Engineering Electrical and Electronics Engineering, Vol. 13, Issue 13, Ver1.0, pp. 34-41, 2013.
18. Manuj Darbari, Rishi Asthana and Vivek Kr. Singh, 'Integrating Fuzzy MDE-AT Framework for Urban Traffic Simulation', International Journal of Software Engineering (IJSE), Vol. 1, Issue 2, pp. 24-31.
19. Anneke G. Kleppe and Jos Warmer, 'The Object Constraint Language: Precise Modeling with UML (Addision-Wesley Object Technology Series)'.

BIOGRAPHY

Ravi Pratap Singh has received his B. Tech degree in Information Technology from United College of Engineering and Research affiliated to UPTU, Lucknow (U.P.) and currently pursuing M. Tech degree in Computer Science and Engineering - Software Engineering at BBD University, Lucknow in 2014-2016.

Dr. Vivek Kr. Singh is currently working as Professor in the Department of Computer Science and Engineering at School of Engineering, BBDU, Lucknow. He is HR Head of BBD Group, Lucknow. He has over 14 years of teaching experience Computer Science & Engineering field. He received the B. Tech degree in Computer Science & Engineering from Purvanchal University in 2001, M. Tech from U.P. Technical University, Lucknow in 2006 and Ph.D. from Shobhit University, Meerut. He has published numbers of papers in referred national journals. His teaching areas are: Theory of Automata & Formal Language, Design & Analysis of Algorithm and Computer Architecture.