Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

Modeling of Multiagent Based Railway System using BDI Logic

Ankit Bhardwaj Supriyo Ghosh Animesh Dutta

***Abstract*—Multi-Agent based modeling & simulation is an evolving paradigm for solving real life problems for solving real life problems for the last one decade. In this paper, we model the manually controlled Indian Railway to a multi agent based automated system which is robust in nature and provide a fully automated collision avoidance guarantee. We consider individual entity such as station, junction and train as an agent and the whole railway network is framed as a graph. To sharpen the solutions of existing system´s problems such as collisions and delays, BDI agent with first order predicate logic is incorporated in this paper. The communication and coordination between respective agents (with help of their BDI and rule base running as backbone) can now be a substitution in place of manual decisions. Thus the idea can replace the hectic job of railway control room personals in more sophisticated & methodical way.**

***Keywords*—Agent, Multiagent Technology, BDI Logic, Predicate Logic, Railway Management, Distributed System.**

I. Introduction

Railway system is the biggest public transportation in the world which is used to provide better transportation service to the public than private one. There are lots of advantages of using the public transport such as lesser fuel consumption, convenience, lesser fare, lesser pollution and many more. But in parallel with these advantages the existing railway system has some disadvantages also like Collisions and delays and unfortunately they are more dangerous and lossy than private transport. The probability of errors becomes higher due to involvement of human being in decision taking in technical operations of railway system. A large part of railway system is dependent on human reliability and manual operations. Human performance may be affected by many factors such as age, state of mind, physical health, attitude, emotions etc. Apart from the human reliability, the existing railway system has some other problems also. One of them is fixed Signaling System. Dependency on signaling system introduce some new challenges and problems and some of them are

- Train driver has to watch out for signal board while driving in day or night, even in bad weather also.
- Possibilities of reading aspect of any adjacent signal not meant for his route may lead to accidents.
- Driver has to control his train as per the aspect seen till he reaches / passes the signal.
- The Signals cannot convey and temporary speed restriction imposed on account of any reasons.

Ankit Bhardwaj, Supriyo Ghosh, Animesh Dutta
Department of Information Technology,
Nation Institute of Technology Durgapur,
India - 713209

Railway is a distributed system which can be efficiently modeled by the intelligent agents. A Software Agent [1] is essentially a special software component that has autonomy to provide an inter-operable interface to an arbitrary system. A Multiagent system (MAS) [2] is a group of agents, which interact with each other and negotiate in a decision to achieve a common goal. Multiagent system based modeling and simulation has been already done in various problem domains [3], [4], [5], [6]. MAS can be empowered with the help of BDI theory. BDI is a philosophical and mental model for multiagent based problems. Railway system is a Socio-Technical system which can be modeled by BDI agent [7] using three words Belief, Desire and Intension. During his lifetime an agent go though in following working loop depicted in Fig:1 :

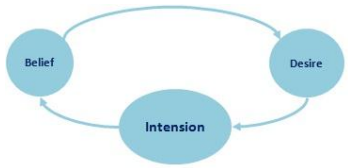

Fig. 1: Working principle of BDI

Now apart from the BDI theory we are going to concatenate first order predicate logic in this paper which empowers the railway model many times. The combination of mathematical logic and BDI theory gives stronger solution for railway system problems because the logic gives precise meaning to mathematical statements.

The remaining section of the paper is structured as follows: in section II & III the Related work and Scope of work is described, section IV depicts the System model. Section V deals with problem formulation & their solution. In section VI a Case study is given. Finally we draw a Conclusion and future direction of the work in section VII.

II. Related Work

A number of papers have been presented for the intelligent transportation system. In [8], author works for train punctuality and he uses GSM-R to collect the information about other trains such as speed and time needed to finish the journey and on board processors to adjust speed of the train automatically. In papers [9], [10] authors give a satellite based controlling for railway system which is based on GPS (Global Positioning System) technology. In [9] author proposed to store the position of the trains located by GPS into a web server so a train can get position of the other trains from web server. In paper [10] a

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

concept of on board GPS receivers is suggested for controlling the Indian railway system. They state that GPS receiver can locate the position of trains and stations and then driver can take decision accordingly. Both of these papers suggest a centralized system for controlling trains and failure of satellite or web server can breakdown the whole system. Authors in [11] overcome the problems of centralized system by introducing a innovative idea of distributed multi agent system in transport. They concern the application of distributed multi agent for presenting an appropriate model of interacted distributed layers for railway transportation system but this paper is lacking to give any solution for problems that are facing in existing system. Paper [12] is presented for learning model of agents from human behavior and gives some rule based model for vehicle controlling. This paper proposed an idea to enhance agent reasoning. Authors in [13] propose an agent based modeling for railway system using fuzzy logic. Paper [14] also propose a multi agent based railway system and that system is subdivided into two different subsystems (i) Control (ii) Learning. Both the papers are limited to modeling of the system and does not provide any solution for actual problems related to collisions and delays in the system.

In paper [15] authors present first-order temporal BDI logic for forest multi-agent system and give a hierarchy of the agents and define some rules for agents. They give a case study and simulation for hierarchical agents also. This paper is lacking to give logics for interaction among sibling agents.

In [16] authors give a sound BDI architecture for autonomous locomotive control and they provide working model of multiagent system using BDI. Main attention of the authors is towards the acceleration and deacceleration of the locomotive only. It does not meet all requirements to solve some common problems of railway system such as train passing, controlling single edge track between two stations etc.

III. Scope of Work

Railway plays a vital role in growth and economy of a country by giving better connectivity among the different parts of the nation. Demand of railway services are increasing rapidly with growing population. To make equilibrium between demand and supply of railway services, some advance technologies are required to enhance the quality of service. Though a lot of novel works have been done to improve the punctuality of trains, it is not adequate for the purpose. These systems need a human intervention and a periodic conscious observation. So a scarce revolution is required in this domain which can automate the whole system successfully. In our previous work [17] we design a mechanism to automate the railway system using agent technology emphasizing in syntactic communication only. To achieve a better quality of services we propose an Agent based model using BDI Logic for railway system which can act autonomously. BDI Agent modeling can replace the problem of fixed signaling approach as there is an autonomous inter-agent communication in our system. We deploy an agent to every train, station and junction to take their own decision and construct individual agent´s BDI which can give a valid flavor of the mental model of drivers. A rule base is also running as a backbone of the system, which can help the agent to take their autonomous decision. Finally our proposed system must guarantee a collision free environment which can accommodate various high speed trains with minimum delay.

IV. System Model

We divide the modeling of railway system in two phases. In first phase we give a graph representation for tracks and stations and indicate agents present in the system. In second phase we define the BDI architecture for these agents.

A. *System Architecture*

We define railway system by following tuples set:

$$R_{sys} = < G, A>$$

Where, G is a Graph representation of railway tracks and stations.

A is a set of agents deployed on the graph.(Figure: 2)

Now the graph G can be defined as

$$G = < V, E >$$

Where, V is the set of vertices and E is the set of directed edges that connect two adjacent vertices of the graph. G.

Again, $V = S \cup J$

$S = \{ S_i \mid i=1,2\ldots.n\}$

$S_i \epsilon S$ represents a station node with an unique id i

n= Total number of stations in the system.

And $J = \{ J_i \mid i=1,2\ldots.m\}$

$J_i \epsilon J$ represents a junction node with an unique id i

m= Total number of Junctions in the system.

Main characteristics of a junction point are

- Maximum 1 train can pass at a time from it.
- No train can wait on the junction point.

$$J = \{ x \mid x \epsilon V, \text{and } Capacity(x)=1 \text{ and } Waiting\_on(x)=0 \}$$

Where, *Capacity(x)* = Maximum number of platforms at x and *Waiting_On(x)* = Waiting time of train on station x.

In the graph of this system model we are considering the directed edges so trains can move in these directions only .

$$E_{ijk} = V_i \rightarrow V_j$$

$$And \quad E = \{ E_{ijk} \mid V_j = Adj_G(V_i) \}$$

Here $Adj_G(V_i)$ is a adjacency function which determines the adjacent nodes of V_i .

As there may be more than one edge between V_i to V_j , to uniquely define each edge between V_i to V_j we define E_{ij} as E_{ijk}. Where $\{ k = 1, 2....e \}$ and e is the total number of edges directed from V_i to V_j .

Now the another parameter of the system model is A (agent),

$$A = SA \cup JA \cup TA$$

Where $SA = \{ SA_i \mid i=1,2....n\}$

$SA_i \epsilon SA$ represents a Station Agent with an unique id i which is static in nature.

$$JA = \{ JA_i \mid i=1,2....m\}$$

$JA_i \epsilon JA$ represents a Junction Agent with an unique id i which is static in nature.

According to model:

$$\Phi : SA \rightarrow S$$

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

$$\Phi : JA \rightarrow J$$

Φ is a function which deploys an agent $SA_i \in SA$ into a station node $Si \in S$. Φ also maps a agent $JA_i \in JA$ into a Junction point $J_i \in J$ and keeps the information of J_i.

$$TA = \{ TA_i \mid i=1,2....N\}$$

Where, $TA_i \in TA$ symbolize a Train Agent with an unique id i which can travel in the graph G with a predefine strategy.
N= Total number of trains in the system.

$$\Phi_M: TA \rightarrow (V \cup E)$$

Φ_M is a Mapping function that a mobile agent $TA_i \in TA$ at any time instance t should be deployed on any point of the graph G. In our system model a train agent can be defined by the following attributes,

$$TA_i : \textbf{<ID,R,ST,WT,MPS>}$$

Where, ***ID*** = Unique id of the train agent TA_i. ***R[i][j]*** is List of stations that TA_i must follow as its Route and $\{ j = 1, 2.... n \}$, while n is the total number of stations in the route of TA_i. ***ST[i][j]*** is Schedule time of TA_i in the j^{th} station (R[i][j]),***WT[i][j]*** is Waiting time of TA_i in the j^{th} station (R[i][j]), ***MPS*** is Maximum permissible speed of TA_i.
Similarly SA_j: ***<ID,Capacity,LT,AS,Edge>***
And JA_k : ***<ID,LT,AS,Edge,B>***
Where, ***ID*** is Unique id of the respective agent, ***Capacity*** is maximum number of trains SA_j can hold at a time, $LT[p]$ is List of trains those left SA_j or JA_k but still not reached to the next stop, $AS[p]$ is List of adjacent stations of SA_j or JA_k, $Edge[p]$ is List of outgoing edges of SA_i, and B is the boolean status of junction point.

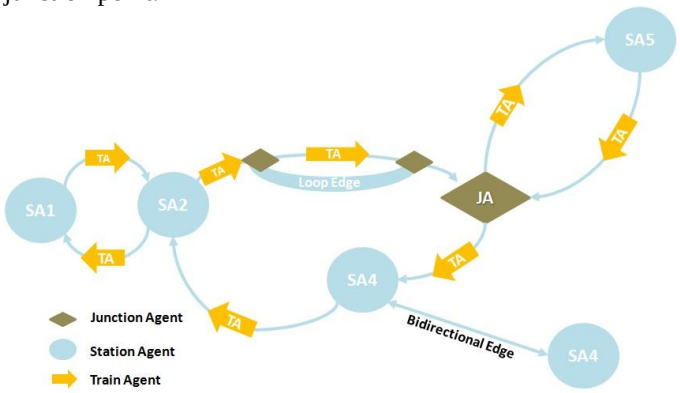

Fig 2 : System Model

B. *BDI Architecture of the System*

To differentiate among tags of BDI, we give the following well defined structure to represent belief, desire and intension. The predicates and functions which we use frequently to define the BDI of a agent are summarized in Table: I.

1) Belief Logic Tags: $<B\ A\ Belief_1>$

Here < > is used to denote the tags. First character of tag can have three possible characters *B*, *D* and *I* which denote the tag type. *B* is used for Belief tags. Second character of the tag contains agent id. In above belief tag, we put the agent id *A*, who has the state of environment $Belief_1$ as belief.

2) Desire Logic Tags: $<D\ A\ Desire_1>$

Similarly, here *D* denotes the tag type that is desire and *A* denotes the agent id. The above tag denotes that agent *A* has desire $Desire_1$. Like $<D\ TA_1\ CurrentSpeed(MPS)>$ tag denotes that *TA1* has a desire to gain the *MPS*.

3) Intension Logic Tags: $<I\ A\ Intension_1>$

Intension is a desire with commitment to fulfill it. So the difference between the Intension and desire tag is tag type symbol only. Rest parts of the tag remain same for an intension tag of a agent like desire tag.

Table I: Predicates used by Agents

Predicates	Description
CurrentSpeed(x)	Current speed of train agent is x
CurrentEdge(x)	Current running edge of train agent is x
TrainPriority(x)	Priority of train agent x
InFront(x)	Train x is in front of a train agent
CriticalDistance(x, y)	Agent x and y are at critical distance
PassRequest(x)	Pass request is to be send to agent x
PassRequestReceived(x)	Train agent has received a request to pass from train x
HasPassRequest(x)	Train agent has a request to pass from train agent x
Accelerate(x)	Accelerate train agent x
Deaccelerate(x)	Deaccelerate train agent x
SendPassRequest(x,y)	Train agent x sends a request to pass to agent y
NextStop(x)	Train agent has next stoppage at x
NextStation(x)	Train agent has next arrival station x
NextEdge(x)	At track path of train, next coming edge is x
PlatformRequests(x)	Station has x number of requests for platform
SendEdgeRequest(e, x)	Train/Station agent sends a request to agent x for edge e
BidirectionalEdge(x)	x is a bidirectional edge
StopFlag(True/False)	Flag set by Station/Junction to Deny/Allow a train to enter
Payoff(x,pay)	Train agent x has payoff value pay
PassFlag(True/False)	Flag set by train agent to Allow/Deny for pass
LoopEdge(x)	x is a loop edge
PlatformCapacity(x)	x Platforms are free at a station

v. Problem Formulation and Solution

Problem 1 : If two trains or mobile agents are running on the same edge (obviously in the same direction) and relative distance between them becomes less than a critical distance, then probability of collision becomes very high.

Solution 1: Two train agents TA_1 and TA_2 are moving at same edge e in same direction and TA_2 is rear train. We take $TrainPriority(TA_2) > TrainPriority(TA_1)$ and *MPS* of TA_2 higher than *MPS* of TA_1 as assumptions. First assumption means when TA_2 comes at critical distance with TA_1, TA_2 deaccelerates upto speed of TA_1 and TA_1 should provide a pass

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

to TA_2 on request. To achieve this solution, TA_1 and TA_2 will follow the Algorithm: 1.

if $\langle B\ TA_1\ CurrentSpeed(sp1) \wedge CurrentEdge(e)\rangle$
then
  $\langle D\ TA_1\ CurrentSpeed(mps)\rangle$;
end
if $\langle B\ TA_2\ CurrentSpeed(sp2) \wedge CurrentEdge(e) \wedge InFront(TA_1) \wedge CriticalDistance(TA_2, TA_1)\rangle$ **then**
  $\langle D\ TA_2\ CurrentSpeed(mps)\rangle$;
  $\langle D\ TA_2\ PassRequest(TA_1)\rangle$;
end
//Now these desires will become the Intension for the system ;
if $\langle I\ TA_1\ CurrentSpeed(mps)\rangle$ **then**
  $Accelerate(TA_1)$;
  $Untill\langle B\ TA_1\ CurrentSpeed(mps)\rangle$;
end
if $\langle I\ TA_2\ CurrentSpeed(CurrentSpeed(TA_1)) \wedge PassRequest(TA_1)\rangle$ **then**
  $Deaccelerate(TA_2)$
  $Untill\langle B\ TA_2\ CurrentSpeed(CurrentSpeed(TA_1)) \wedge \sim CriticalDistance(TA_2, TA_1)\rangle$;
  if $\langle B\ TA_2\ TrainPriority(TA_2) \geq TrainPriority(TA_1)\rangle$ **then**
    $SendPassRequest(TA_2, TA_1)$;
    $Untill\langle B\ TA_2\ \langle B\ TA_1\ PassRequestReceived(TA_2)\rangle\rangle$;
  end
end
while $\langle B\ TA_2\ InFront(TA_1)\rangle$ **do**
  if $\langle B\ TA_2\ PassFlag(True) \wedge \sim CriticalDistance(TA_2, TA_1)\rangle$ **then**
    $\langle D\ TA_2\ CurrentSpeed(mps)\rangle$;
  end
  if $\langle I\ TA_2\ CurrentSpeed(mps)\rangle$ **then**
    Accelerate(TA_2);
    $Untill\langle B\ TA_2\ CurrentSpeed(mps)\rangle \wedge \langle B\ TA_1\ InFront(TA_2)\rangle$;
  end
end

Algorithm 1 : Maintain Critical Distance and Passing of Train(TA_1, TA_2, *Edge*)

Problem 2: Every station Si 2 S has a predefined capacity (Maximum number of trains it can hold at any time instance). So, at time t if all the platforms of a station is occupied and a train wants to enter, a collision may occur.

Solution 2: Train agent TA wants to enter in station SA and demand a platform from SA to stay. We are taking a Platform flag which is set by SA only. This flag has two values: True for permission grant to train to acquire platform and False for deny. Flag will be set True by SA if a Platform is available. Algorithm: 2 gives solution of platform allotment.

if $\langle B\ TA\ CurrentSpeed(sp) \wedge NextStop(SA) \wedge NextStation(SA) \wedge CriticalDistance(TA, SA)\rangle$ **then**
  $\langle D\ TA\ GetPlateform(SA)\rangle$;
  $\langle D\ TA\ CurrentSpeed(0)\rangle$;
end
if $\langle B\ SA\ PlatformCapacity(\sim Full) \wedge PlatformRequests(\sim Null)\rangle$ **then**
  $\langle D\ SA\ PlatformFlag(True)\rangle$;
end
while $\langle I\ TA\ GetPlatform(SA)\rangle$ **do**
  if $\langle I\ SA\ PlatformFlag(True)\rangle$ **then**
    $SetPlatformFlag(True)$;
  else
    $\langle D\ SA\ PlatformFlag(True)\rangle$;
  end
  if $\langle I\ TA\ GetPlatform(SA)\rangle$ **then**
    if $\langle B\ TA\ PlatformFlag(True)\rangle$ **then**
      GetPlatform(SA);
      $\langle B\ TA\ GotPlatform(SA)\rangle$;
    else
      Deaccelerate(TA) ;
      Untill $\langle B\ TA\ CurrentSpeed(0)\rangle$;
    end
  end
end

Algorithm 2 : Platform Allotment (*TA, SA*)

Fig 3: Railway model with Loop Edge

Problem 3: Whenever a train enters in a Junction point where it should not stop and in its outgoing edge another train is waiting within the Critical Distance, then a collision may occur.

Solution 3: A Junction agent presents at any meeting point of more than two edges. Here we are assuming a junction point in between loop edge le and normal edge shown in figure 3. Algorithm: 3 will be followed to solve this problem.

Problem 4: A bidirectional edge exists in between two stations SA_1 and SA_2 or vertices V_i and V_j then collision may happen if two trains are moving in opposite direction.

Solution 4: SA_1 and SA_2 have only edge in between them is e and it is bidirectional so only one train can use this edge at a

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

time. Here we are assuming that SA_1 is request sending initiator. Train TA_1 is waiting at SA_1 with payoff value *pay1* and train TA_2 is waiting at SA_2 with payoff value *pay2*. After negotiation in between SA_1 and SA_2, the edge will be allowed to use by the train who has higher payoff value. Algorithm: 4 will come in light to solve this problem.

```
if ⟨B TA1 CurrentEdge(e) ∧ NextJunction(JA) ∧
HasPassRequest(TA2) ∧ ⟨B JA LoopEdge(le)⟩⟩
then
    ⟨D TA1 GetLoopEdge(le)⟩ ;
end
if ⟨I TA1 GetLoopEdge(le)⟩ then
    if
    ⟨B JA LoopEdge(le) ∧ LoopEdgeStatus(Free)⟩
    then
        ⟨I JA SetLoopFlag(True)⟩ ;
        Train TA1 will GetLoopEdge(le) ;
    else
        ⟨I JA SetLoopFlag(False)⟩ ;
        Stop Train TA1 ;
    end
end
```

Algorithm 3: Junction at Loop Edge (TA_1, JA)

```
if ⟨B SA1⟨B TA1 NextEdge(e)⟩ ∧
BidirectionalEdge(e) ∧ Payoff(TA1, pay1)⟩ then
    ⟨D SA1 SendEdgeRequest(e, SA2)⟩ ;
end
if ⟨B SA2⟨B TA2 NextEdge(e)⟩ ∧
BidirectionalEdge(e) ∧ Payoff(TA2, pay2)⟩ then
    ⟨D SA2 SendEdgeRequest(e, SA1)⟩ ;
end
if ⟨I SA1 SendEdgeRequest(e, SA2)⟩ then
    SendEdgeRequest(e, SA2) ;
    if ⟨B SA2 Payoff(TA1, pay1) >
    Payoff(TA2, pay2)⟩ then
        ⟨D SA2 StopFlag(False)⟩ ;
        if ⟨I SA2 StopFlag(False)⟩ then
            SetStopFlag = False ;
            //Permission granted to TA1 to use edge e
        end
    else
        ⟨D SA2 StopFlag(True)⟩ ;
        if ⟨I SA2 StopFlag(True)⟩ then
            SetStopFlag = True ;
            //Permission denied to TA1 to use edge e
        end
    end
end
```

Algorithm 4: Bidirectional Edge Negotiation (SA1, SA_2, e)

VI. Case Study

For the application of our BDI predicate logic, we take an example railway scenario (Figure:3) in which we take two station and two train agents who have unique ids and these are SA_1, SA_2, TA_1, TA_2 for station1, station2, train1, train2 respectively. These agents have their own beliefs, desires and intentions and decide the action plan accordingly. To use our proposed algorithms in this scenario, we classify the scenario in sub scenarios as When two trains are at same edge (trains are moving in same direction) and within critical distance and another When a train is entering into a station.

Train TA_1 has left from SA_1 at time t_1 and after Δt interval of time, train TA_2 also left from SA_1 at same edge The train priority of TA_2 is higher than train TA_1 and *MPS* of TA_2 is also higher than TA_1. So we can say that after a certain time interval TA_1 and TA_2 will be at critical distance. TA_1 and TA_2 are in communication and rear train TA_2 follows the Algorithm: 1 to avoid the collision. The overall idea of the Algorithm: 1 can be shown by the following predicate logics.

$\forall x \forall y$ *CriticalDistance(y,x)* $\wedge$ *InFront(x)* $\rightarrow$ *Deaccelerate(y)*

$\forall x \forall y$ *CriticalDistance(y,x)* $\wedge$ *InFront(x)* $\wedge$ *HighPriority(y)* $\rightarrow$ *Deaccelerate(y)* $\wedge$ *PassRequest(y,x)*

where x, y represents the trains.

After successful execution of Algorithm: 1 the train TA_2 is in front of TA_1 and it moves towards the next station SA_2. Now the problem of platform allotment comes in the light. After coming at a critical distance with SA_2, the message passing takes place in between TA_2 & SA_2 and Algorithm 2 is used. The platform is acquired by TA_2 if platform flag is *True*. Platform flag is set *True* or *False* by station agent after checking available platform capacity. Algorithm: 2 can be concluded by the following logic.

$\forall t \ \forall s$ *CriticalDistance(t, s)* $\rightarrow$ *Deaccelerate(t)* $\wedge$ *PlateformRequest(t,s)*

Where *t, s* represent train and station respectively. After getting a positive response from station SA_2, train TA_2 acquires the platform which has flag value as *True*.

VII. Conclusion

In this paper we propose a BDI logic based modeling of railway system which gives a strong foundation to solve the problems of existing railway system and also gives an autonomous shape to it. Our proposed model promotes the utilization of predicate logics with multi agent system to enhance the quality of service of the system. In a concise form we can state that this paper is directed to develop a BDI agent based autonomous railway system (with the rule base as a backbone) which can overcome the fallacy of manual intervention and human dependency.

The future prospect of this work will be simulate the whole network with JADE (which is a Java based agent oriented tool) by using the real dataset collected from the ASANSOL division of Eastern Railways, India which can give the flavor of real

ISBN: 978-981-07-7021-1

time simulation. Then we want to deploy our system in a real physical environment to achieve our final.

0

References

[1] M. Wooldridge and N. R. Jennings, Intelligent Agents: Theory and Practice, The Knowledge Engineering Review. vol. 10, 1995, pp. 115-152.

[2] M. J. Woolbridge, Introduction to multiagent system, John Wiley Sons,Inc(2001).

[3] M. Jacyno, S. Bullock, M. Luck, and T. Payne, Emergent service provisioning and demand estimation through self-organizing agent communities, Proc. 8th Int. Joint Conf. AAMAS, 2009, pp. 481-488.

[4] L. S. Tesfatsion, Introduction to the special issue on agent-based computational Economics J. Econ. Dyn. Control, vol. 25, no. 3-4, pp. 281-293,Mar. 2001.

[5] S. M. Lee and A. R. Pritchett, Predicting interactions between agents in agent-based modeling and simulation of sociotechnical systems, IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 38, no. 6, pp. 1210-1220, Nov. 2008.

[6] M. Vasirani and S. Ossowski, A market-inspired approach to reservation based urban road traffic management, in Proc. 8th Int. Joint Conf. AAMAS, 2009, pp. 617-624.

[7] AS. Rao and M.P. Georgeff, BDI Agents: From Theory to Practice, Proceedings of the First International Conference in Multi Agent Systems (ICMAS-95), San Francisco, USA, 1995.

[8] Ingo A. Hansen, Improving railway punctuality by automatic piloting, 2001 IEEE Intelligent Transportation Systems Conference Proceedings - Oakland (CA) USA - August 25-29, 2001.

[9] N. K. Das, C. K. Das, Rajesh Mozumder, Jiban Chandra Bhowmik, Satelite based Train Monitoring System, Journal of Electrical Engineering The Institution of Engineers, Bangladesh Vol. EE 36, No. II, December 2009.

[10] S.K.Biswas, GPS BASED CAB-SIGNALLING FOR INDIAN RAILWAYS - S.K.Biswas, GPS BASED CAB-SIGNALLING FOR INDIAN RAILWAYS

[11] Ali Pouyan, Momeneh Taban, Sadegh Ekrami, A Distributed Multi Agent Control Model for Railway Transportation System, ICAS 2011 : The Seventh International Conference on Autonomic and Autonomous Systems.

[12] Hiromitsu Hattori, Yuu Nakajima, and Toru Ishida, Learning From Humans: Agent Modeling With Individual Human Behaviors, paper appears in Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on Jan. 2011.

[13] Ali Siahvashi, Bijan Moaveni, Automatic Train Control based on the Multi-Agent Control of Cooperative Systems, The Journal of Mathematics and Computer Science Vol .1 No.4 (2010) 247-257.

[14] Hugo Proenc, Eugenio Oliveira, Multi-Agent Railway Control System, Published in 9th Lbera-American Conference on AI , Puebla, Maxico, November 2004.

[15] Lijun Wu, Kaile Su, Abdul Sattar, Qingliang Chen , Jinshu Su, Wei Wu, A complete first-order temporal BDI logic for forest multi-agent systems, Published in Journal Knowledge-Based Systems Volume 27, March, 2012Pages 343-351.

[16] Marcos R. da Silva, Andre P. Borges, Osmar B. Dordal, Denise M. V. Sato, Bniulio C. Avila, Fabrfcio Enembreck, Edson E. Scalabrin, An Architecture of BDI Agent for Autonomous Locomotives Controller, Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design.

[17] S. Ghosh and A. Dutta, Multi-agent based railway track management System, Proceedings of the 3rd IEEE International Conference on Advance Computing & Communication(IACC-2013), Ghaziabad, India, Feb. 2013.

About Author (s):

Ankit Bhardwaj is currently pursuing his M.Tech degree from NIT Durgapur India in Information Technology. He received his B.Tech degree from Uttar Pradesh Technical University, India in Computer Science & Engineering. His research interests are Multi-agent system, Social environment simulation and modeling the mental model of Agent.

Supriyo Ghosh is pursuing his M.tech degree from NIT Durgapur India in Information Technology. He received his B.Tech degree from West Bengal University of Technology, India in Computer Science & Engineering in 2011. His research interests are Multiagent based social environment simulation, Intelligent Agent decision making, interaction between human & computer etc.

Animesh Dutta received his B.Tech and M.Tech degree from NIT Durgapur India in Computer Science & Engineering in 2002 & 2004 respectively. He joined Information Technology department, in 2008 as a Lecturer. He is currently serving as Assistant Professor there. His research interests include Distributed System, Multiagent System, Software Engineering etc.