TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics and Mechanics
Vol. 54, Issue II, 2011

RAILWAY TRAFFIC MANAGEMENT IMPLEMENTED BY A MULTI-AGENT SYSTEM

Maria-Magdalena SANTA, Octavian CUIBUS, Tiberiu LEȚIA and Mirela ENACHE

Abstract: The problem of railway traffic management is very complex one involving major outcomes in a modern society. Trains are suitable for transporting peoples and goods with a good trade-off between cost and rapidity. The increase of railway traffic efficiency and flexibility requires new methods for management. New charter trains have to be added continuously without disturbing the other periodic train moves or decreasing the safety conditions. A railway system consists of a network of tracks, list of stations, safety devices (signals, sensors, etc) and a set of trains. Each train has assigned a corresponding train agent. Distributed applications are applicable in case of large systems like railway network. This paper presents a model for railway network management that operates in dynamic environments. The paper also provides a distributed management algorithm based on train agent negotiation. Dynamic attributes of algorithm include those that are measured and evaluated online.
Key words: *traffic railway management, multi-agent system*

1. INTRODUCTION

Train movement control and management is a complicated activity even with computer support. Trains are suitable for transporting peoples and goods with a good trade-off between cost and rapidity.
A railway system consists of a network of tracks, list of stations, safety devices (signals, sensors, etc) and a set of trains. Trains moves from one station to other along the network. The trains must be directed to the correct destinations within a tight time schedule and the physical limitations of the track network impose substantial constraints on train movement. Obviously, two trains travelling on the same track cannot pass each other in opposite directions or in the same direction, except where sidings occur.

The railway control system is a dynamic one that operates in an environment with uncertain, properties that include transient and resource overloads, arbitrary arrivals, arbitrary failures and decreases of traffic parameters. Once a delayed train deviates from its original time-distance path, it may conflict in passing or meetings with other trains. Despite many uncertainties, the control system expected to guaranty that all the trains behave according to timelines. For solving very large railway simulation problems a distributed architecture approach must considered. The main reasons for this are scalability, performance and reliability.
The railway traffic management problem can, therefore, be defined as follows: given a railway network, a set of train routes and passing/stopping times at each relevant point in the network, find a deadlock free and conflict free schedule with no unsolved overlaps between the blocking times of trains, compatible with their initial positions and such that the selected train routes are not blocked, the speed profiles are acceptable, no train appears in the network before its intended entrance time, no train departs from a relevant point before its scheduled departure time, rolling stock constraints and connected train services are respected and trains arrive at the

relevant points with the smallest possible delay. Using Global Positioning System (GPS) and the wireless communication some moving block sections can be implemented. This can lead to higher track utilization, but traffic safety is based on GPS and wireless communication system reliability.

The important basic idea is “agent”, it means an intelligent entity with some characters and functions such as: autonomy, activity, reactivity, mobility.

Multiagent systems are an approach to build complex distributed applications. A multiagent system consists of a population of autonomous agents situated in a shared structured system. For solving distributed problems are involving train agents with some degrees of autonomy and rationality capability. This field of study contains models for both decision-making and negotiation activities between trading parties. As a result, it provides a potentially viable algorithm to modelling the transaction between train agents.

The traffic system goals are: to minimize traffic cost, to maximize traffic system throughput, to fulfil train timing requirements, to guaranty system safety, to minimize fault effects on train schedules and to sustain railway maintenance.

2. RELATED WORK

The basic principles of railway traffic control are given in [1]. These include the interlocking usage, resource management and dividing the railway network into different parts. The assessment of scheduling is performed by the capability of the schedule to meet the needs of customers and the capabilities of the trains to recover the delays according to their timetables. The train deviations from the scheduled timetable should be removed during the operation [2]. New trends of train traffic control and management are started since 1997 by [3]. An autonomous decentralized train control and management system is proposed to attain both the real-time properties for train control such as the real-time traffic and non-real-time properties for train management. A single delayed train can cause a domino effect of secondary delays over the entire network, which is the main concern of planners and dispatchers, [4].

Train scheduling implementations are:

-offline scheduling when all the train arrival times and departure times are calculated before the train starts. The trains behave exactly as they were planned. No unexpected event happens and no new train can appear.

-online scheduling when the scheduling is performed during the train traffic operation. Some trains have variable delays, unexpected events happen, and new train scheduling requests are required and accepted during the operation.

Formal development and verification of a distributed railway control system are performed applying a series of refinement and verification steps [5]. The distributed train scheduling problem has some similarities with distributed software job scheduling [6],[7]. Both have to fulfil real-time constraints relative to finishing time, communication requests and resource management.

The problem of the railway interlocking scheduling has some common features with independent scheduling of each node of a distributed software system. Each node constructs its local schedule using only local information. The lack of global information makes it impossible for a node to make a globally optimal decision. Thus it is possible for a node to make a scheduling decision that is locally optimal in terms of the utility that can be accrued to the node, but compromises global optimality.

The approach in [8] targets a wide class of control systems (linear systems). The control performance optimization is based on feedback information from the controlled systems, and it is achieved by dynamically adjusting the sampling period of each controller, which is determined at each resource re-allocation. The optimal resource allocation policy for control tasks that they present in this paper optimally solves the Quality of Control (QoC) scheduling problem formulated by [9], but in terms of resource management.

The dynamics of the controlled systems as suggested by [10]and [11] are the key to better exploiting system resources and improving control systems performance in resource constrained control systems. In [11] presents a control-based model for control tasks that allows each control task to trigger itself optimizing computing resources and control performance. Using this model, at each control task instance execution, the executing instance informs the scheduler when the next instance should be executed. The next instance execution point in time is dynamically obtained as a function of the utilization factor and control performance. Preliminary results of the feedback approach to resource management were reported previously by [12]. They present a framework in which a flexible, integrated real-time system directly supports adaptive control applications.

[13] presents a distributed control system contains only local controllers and train agents. The train agent is responsible to get a reserved path from a platform to a critical resource situated on the frontier. It performs the reservation using the local resource table. The train agent requests the complementary train agent (from the neighbour station) to accept the train on the critical resource during a specified period of time.

[14] discusses the applications of agent-oriented techniques for modelling of the behaviour of a railway transportation system which consists of numerous interacting parties with different objectives, interests, autonomy, constraints, responsibilities and decision-making capabilities. The agent model covers both business and operational aspects of the system and it enables possible connections to other modes of transportation and/or external monitoring bodies. The multi-agent architecture required to match a real-world railway system will be presented and the additional functions and advantages brought by the agent model will also be explored.

3. THE TRAIN RAILWAY STRUCTURE

To understand the suitability of the multi-agent concept on railway operation, it is necessary to have a clear picture of the roles of all participants within a railway system and the relationships among them. This chapter describes the physical parts of a railway line. This description is based on a simple railway line with low intensity traffic. Compared to a real railway line many domain specific details have been left out to simplify the model. The railway line is divided into a number of coherent segments. A segment is basically just a division of the railway line. At each end of a segment is a switch box. The segments are used by the control system which, through the switch boxes, controls access to each segment. Figure 1 presents a part of a railway network two stations A and B are connect by two lines L1 and L2. Each station have three platforms, denoted by P1, …, P6. Platforms can form a station.

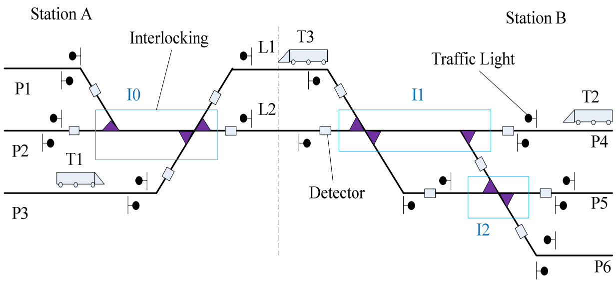

Figure 1. An example of railway structure.

The presence of trains on the line is signalized by detectors, represented by squares. Detectors situated at the border between two segments. When a train passes from one segment to the next, the detector becomes active at the time of the train enters the new segment. The detector becomes inactive again when the train enters the new segment.

Signals (traffic lights) are located before every junction as well as along the lines and inside the stations. Signals added to split the long track lines into smaller block sections (segments) to increase the track utilization. In this case, a safe policy requires that each block section (segment) contains only one train at a time and between each pair of trains a non occupied block section is compulsory.

Interlocking are denoted by I0, I1, I2. An interlocking is an arrangement of neighbour interconnected sets of switch points and traffic light signals such the train movements through them is performed in a proper and safe sequence. The travelling from one interlocking to another is usually free movement. The train crossing an interlocking performs an activity directly controlled by the control system.

A train is a number of connected vehicles that moves on a railway line. The trains are denoted by T1, T2 and T3. While moving along the line the train enters and exits segments along the route. It is assumed that the length of a train is always less than the length of any segment. Therefore a train is either on one or two segments at a time. It is on two segments when it passes from one segment to another.

The control system is able to control train access to different parts of the line. The idea of the control system is to allow only one train at a time at a given segment. The term Task is introduced to define the control problem. It consists of the moves of a train from a given platform (line) of a railway station, to a specified platform (line) of another railway station.

According to the real traffic conditions, the time was introduced, the time moments of departures and arrivals of trains being associated and represent deadline for trains. An example of a synthetic description of such a task is given in Table 1. The first character in the place column specifies the station and after dots, the platform is given. The time unit used in the study is one minute. The columns "time" specifies the desired departure and arrival times.

Table 1. Train table.

Train	Departure		Arrival	
	Place	Time	Place	Time
T1	A.P3	0	B.P6	15
T2	B.P4	6	A.P1	25
T3	B.P5	5	A.P2	22

The objective is to find a way to route the trains such as no deadline missed, and departure and arrival times are as close as possible to the given times.

The scheduling problem consists of a set of trains (with different departure times) that have to get the resources such that they are able to leave the departure places and arrive to destination before their corresponding deadlines. The train scheduling can be performed offline or online.

The offline scheduling involves that all the trains are considered unscheduled and the algorithm searches a feasible solution such that all the trains fulfill the timing requirements. If a feasible solution does not exist, then the lower priority trains can wait on passing tracks to permit the higher priority trains to arrive at destination in time. The trains remained on the side tracks are scheduled (without fulfilling the time requirements) when train density has diminished.

The online scheduling problem considers the case when a train set is scheduled; they are in movement when one or more trains demand to be scheduled. The new schedule demand can be performed in the offline manner (considering all the trains unscheduled) searching a feasible solution for all the trains, or the previous scheduled trains moves according the allocated resources and the algorithm search to allocate the available resources to the new arrived trains without disturbing the routing of previous scheduled trains.

Both approaches use the assumption that the trains can release the tracks at the end of the granted duration. The approach that uses the dynamically resource allocation can lead to better resource utilization, but it is prone to deadlock trains. Only an algorithm that uses additionally strong non-blocking rules [15] can use this approach. A relevant requirement for any solution is to avoid the train deadlocks. Any solution that leads to a deadlock has rejected.

The increased complexity of dynamic open scenarios may prevent the possibility of computing optimal local and global resource allocations within a useful and bounded time, as the optimal level of deliberation varies from situation to situation.

4. A MODEL FOR RAILWAY NETWORK MANAGEMENT

The system management for controlling train movement uses a distributed architecture. Distributed control offers several advantages like a reduction in size, the lower total installed cost, the improved behaviour in the event of a failure. The management of large projects as train movements requires analytical tools for scheduling activities and allocating resources. Figure 2 give the architecture implementation of this method.

The control system for railway system is composed by a number of train agents A1, ..., An, one station server Sa for station A and Sb for station B, respectively, local controller Ca and Cb, resource reservation table (RT) and a communication platform for agent trains.

The trains need to be managed for a path from station A to station B. The server station receives signals from individual agent trains, including position information derived from a navigation system. The server station uses are: global information, local information: state and reservations in resource reservation table.

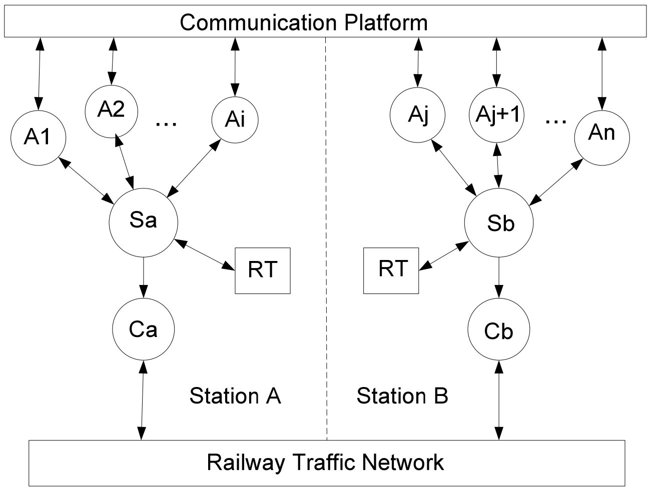

Figure 2. System Architecture.

The local controller assigned to each station has the task to implement the train schedules received from server station. The controllers coordinate local train movement. For this purpose, the controller signals the traffic lights and the interlocking. Each railway route element has associated a min-max interval. Also, each resource has associated a detector that signals the presence or the movement of trains in the corresponding zone. Immediately after a train finishes the crossing of an element, the detector sends a signal to the local controller.

It can be noticed that the delays associated to the transitions of the controller, determining the timed commands sequence, are in accordance with the time intervals for which the resources are occupied.

A train specification refers to the shortest duration needed by a train to cross a resource.

The sequence of a path contains the elements R_i (t_{min},t_{max}) where R_i is the needed resource, t_{min} is the earliest time when the resource is required and t_{max} is the latest time when the resource is reserved.

A train schedule solution specifies the used resources, the occupancy order and the time spent by the train on each resource.

The resources on the frontier are considered critical resources. They are associated to the critical sections of the tasks. A critical section is an element or a sequence of elements of the path crossing the frontier, where if two trains enter, they cannot avoid the deadlock or the collision. L1 and L2 are the critical resources for the given example.

5. THE AGENT TRAIN BEHAVOIR

The train agent moves from station to station carrying train's information. In addition to the information received from its own sensors, each agent also receives information from other agents via communication platform. The train agent uses the train specification, the resource reservation table to get a path that leads the train from the departure platform to a critical frontier resource. That represents a scheduling task that is particular for each train and arrival time. This path is firm reserved but there are trains with delay time or new train charter. In this case, the new train need a new schedule. The train agent is responsible for making resource management decisions by interacting with other train agents and server station for obtaining resources reservation.

The task diagram of this method is presented in Figure 3. A local control subsystem is composed of a controller, a number of train agents, a resource reservation table and a train schedule table. The controller assigned to each station has the task to implement the train schedules received from train agents. For this purpose, the controller signals the traffic lights and the interlocking. It handles signals received from detectors to advance its execution and to release the reserved resources. The train agent has a list of paths (pathList) from the departure platforms of the train (task) to the accessible critical resources that can be used to fulfil the requirements. This path is firmly reserved for the train that have no delay. If the train has accumulated delay, the train agent is looking for a reservation and if it does not find one it waits on resource. This avoids the crash with another train.

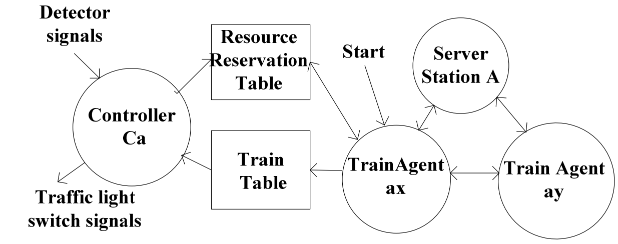

Figure 3. The task diagram of the management method.

For the given example, a train T_x, currently stopped on a platform of a station A and ask a path from train agent, a_x to the destination resource, a platform of the station B. For find a path, train agent a_x negotiations with other train agents, a_i.

Is considered that:

-the trains T_n are scheduled and a new train T_x charter or a train with accumulated delays appears and need a new schedule;

-between station A and B the trains are not breaking and there are not delay time, the trains time are respected;

-the train T_x are moving to the next station if it has all resource need;

-the train T_x have priority (p_x) and for delay the trains are penalized, total delay of a train D_T is sum of delay on each resource: $D_T = \sum_{i=1}^{n} d_i$, where d_i is delay on resource R_i;

-for train's transit each resource R_x has a minim time: t_{min} , a maxim time: t_{max} and a laxity time: t_l , $t_{max} = t_{min} + t_l$;

-every resource R_i has a value $b_i = f(c_i, t_i, p_x)$, where c_i is the resource criticality, t_i is the time need for resource transit, $t_{min} \le t_i \le t_{max}$ and p_x is the priority of train T_x that need the resource;

-each train has specified a laxity time t_l and a deadline to arrive at destination. A train T_x is correctly scheduled and controlled if it arrives at destination before the deadline;

-if a train T_x not respect deadline on a resource R_i is penalized with $q_x = f(p_x, d_i)$ where p_x is the priority of train and d_i is delay time on resource R_i;

-every train T_x has a budget conform to his task and is sum of value resources needed for entire task: $B_{Tx} = \sum_{i=1}^{n} b_i$;

-when a train T_x reserved a resource R_i it spent from budget the corresponding value b_i;

-a train that have a resource can not loose the reservation but it can give the resource in same condition;

-the train T_x with bigger priority p_x or budget B_{T_x} can pay much than a train T_y with smaller priority $p_y \le p_x$ or budget $B_{T_y} \le B_{T_x}$,

-a train T_x with bigger priority p can gives the resource to another train T_y with smaller priority $p_y \le p_x$ if it can respect all deadline after the new scheduler;

-a train T_x with smaller priority p_x can gives the resource to another train T_y with bigger priority $p_y \ge p_x$ if the new scheduler it is a path so that $newB_{T_x} - newD_{T_x} \ge B_{T_x} - D_{T_x}$, means that the total cost is smaller for the new scheduler.

The control decisions take as a target to lead the train such that it reaches the earliest arrival time. The performance evaluation of the system behaviour considers that the control fulfils the requirements if all the trains arrive before their specified latest arrival times. The amount of time needed by any path to complete execution must not exceed the required deadline. The objective is to find an allocation and setting of unknown performance parameter values such that all applications can be scheduled feasibly and the overall utility is maximized.

6. MANAGER ALGORITHM FOR TRAIN AGENT

The train agent is responsible to get a reserved path from a platform to a critical resource situated on the frontier. It performs the reservation using the local resource table conform to the path. Every train T_x has a train agent a_x, a priority p_x and a budget B_x. If a needed resource R_i is occupied, the train agent a_x requests the complementary train agent ay of train T_y, who has the resource reservation, to accept the concession of resource for a price b_i during a specified period of time, (t_{min}, t_{max}). The complementary train agent a_y tries to reserve another path on the resource reservation table. When a request from a train agent arrives, the complementary agent grants or refuses the requirements immediately. This avoids the blocking of the train agent on a particular path reservation. If the reservation succeeds, the complementary agent answers true and the train agent loads the schedule on the train schedule table. Otherwise it answers false. When the answer is false, the train agent cancels the reservation and tries the reservation of another path, another train agent.

Manager algorithm for the train that ask for a scheduler is presented. The train agent is denoted with newTrainAgent.

```
input: resourceTable, trainTable, trainRoute;
initialization: listOfPath = 0;
output:trainScheduler,newResourceTable, newTrainTable;
wait(start); // newTrainAgent ask a path;
while (train is not scheduled)
  do // path reservation
    calculate all paths to listOfPath;
      sort on time listOfPath;
        bestRoute = 1; //select first path on listOfPath
      do for all resource on bestRoute
        // negociation with trainAgent that has reservation on resource
              offer bonification
            trainAgent calculate new path
        trainAgent verify cost of new path
  if   answer = OK return: trainScheduler
else  try with another bonus or resource;
        end do;
      bestRoute ++;
            calculate newResourceTable, newTrainTable
    end do;
```

7. NEGOTIATION

The problem of negotiation is between a buyer and a seller train agent. The buyer agent a_x needs to obtain a resource possessing several attributes (e.g. price, time, etc.), but their exact values will depend on the availability of supply. The train agent therefore proposes a set of criteria based on these attributes. During the negotiation, the buyer train agent has to specify its requirements on these attributes to the seller train agent a_y. When the offer is accepted the message OK is reply. Otherwise, a notOK message is sent and the buyer is prompted. If is modify one of the submitted attributes that corresponds to making a concession. A newOffer message will be issued in some condition and the seller will be asked to generate a new offer. The negotiation is terminated by either an acceptance by OK message or a failure.

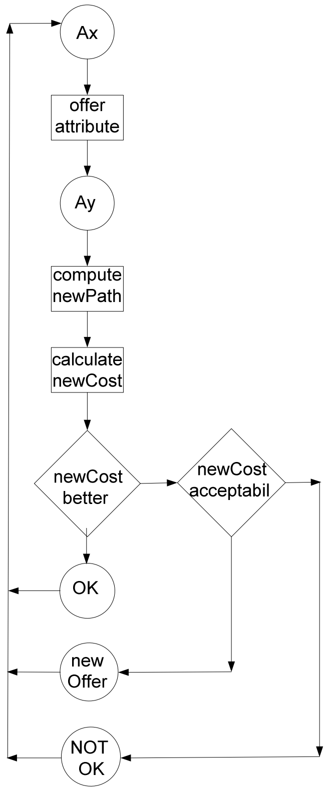

Figure 4. The negotiation algorithm.

The seller train agent a_y is looking for a new scheduler. If it finds one it compute newPath and calculate the cost of the new scheduler and decreases the delays, if there are some. Than if the new cost is smaller, it answers with OK. If the cost is inacceptable, the answer is notOK. If the cost can be accepted with some condition, the message is newOffer.

For a train with big delays or priority, the seller train agent can increase the value for resource and make a new offer.

The role of the seller train agent is to generate a feasible offer according its own benefits. The structure of negotiation algorithm is presented.

In our case, if a train T_x need a resource, the train agent ax ask the train agent ay of train T_y that have the needed resource to accept the concession of resource R_i for a period (t_{min}, t_{max}). For resource R_i train agent ax offer a bonus $b_i = f(c_i, t_i, p_x)$, where c_i is the resource criticality, t_i is the time need for resource

transit, $t_{min} \leq t_i \leq t_{max}$ and p_x is the priority of train T_x.
The priorities of the trains are: freight, the smaller priority, pay for a resource the bonus b on a minute, a train with priority P1, pay the bonus 2b on a minute for the same resource with c_i criticality and a train with priority P2 pay 3b and so on, as in the table 2.

Table 2. Price on time for a resource with c_i criticality.

Priority train	Time (min.)	Price
Freight	1	b
Freight	2	2b
P1	1	2b
P2	1	3b
P3	1	4b
P4	1	5b
P3	2	8b

There are two methods for negotiations:
1. The train agent a that is asking for resource, do not know about the train's reservations of the other train T_y and
2. The train agent a_x that is asking for resource, know the path and the resource reservations of the train T_y and the position of the train.

The second variant is better than first, it avoid transferring not achieved query. In this case, train agents operate with information about sub-tracks that are reserved together. The train agent a_x analyzes if the sub-track needed is reserved by another train and verify if the train could accept the concession of the sub-track. In this case is avoiding the null negotiations.

8. CONCLUSION

The proposed management method does not lead to deadlock due advance resource reservation. It has the advantage to apply on line and so it is able to diminish the variations of the train arrival times. The scheduling algorithm performances can be evaluated compute sum of all trains costs in a period. The proposed method can be used to design the railway networks such that to be capable of providing a specified throughput with real time feature.
This paper highlights the benefits of applying agent technology to rail transportation system modeling. The on-board equipment with some intelligence enhances the autonomy of the train, simplifies the structure and reduces the equipment.
The business and engineering activities in a rail system shown to provide an appropriate platform for agent application. The suitability of agent architecture and structure for system behavior studies has also been discussed. The paper elaborates on the potential for value adding to railway transportation resulting from agent representation. Such an approach can be seen as a catalyst to induce further work on the development of appropriate agent structures for modeling rail transportation and hence to promote the adoption of advance software techniques in railway management and operation.

9. REFERENCE

1. Pachl, J., *Railway Operation and Control*,VTD Rail Publishing, Mountlake Terrace WA 98043 USA, (2004).
2. Tornquist, J. and Persson, J.A., *Train traffic deviation handling using tabu search and simulated annealing*, Proceeding of the 38th Annual Hawaii International Conference on System Science, pp. 73a, (2005).
3. Shoji, S., and Igarashi, A., *New trends of train control and management systems with real-time and non-real-time properties,* Proceedings of the 3rd International Symposium on Autonomous Decentralized Systems (ISADS'97), pp.319-326, (1997).
4. Goverde, R. M. P., *A delay propagation algorithm for large-scale scheduled rail traffic*, Preprints of 11th IFAC Symposium on Control in Transportation Systems, Delft, Netherlands, pp. 169-175, (2006).
5. Hauxthausen A., and Peleska, J., *Formal development and verification of a distributed railway control system*, IEEE Trans. on Software Engineering, Vol. 26, No. 8, pp. 687-701, (2000).
6. Fahmy, S. F., Ravindran, B., and Jensen, E. D., *On collaborative scheduling of distributable real-time threads in dynamic, Networked Embedded Systems*, Proceedings of the 11th IEEE Symposium on Object

Oriented Real-Time Distributed Computing (ISORC), pp. 485-491, (2008).

7. Fahmy, S. F., Ravindran, B., and Jensen, E. D., *Scheduling distributable real-time threads in the presence of crash failures and message losses*, ACM SAC, Track on Real-Time Systems, (2008).
8. Martí, P., Lin, C., Brandt, S.A., Velasco, M. and Fuertes, J.M., *Optimal State Feedback Based Resource Allocation for Resource- Constrained Control Tasks*, The 25th IEEE International Real-Time Systems Symposium (RTSS'04), pp.161-172, (2004).
9. Martí, P., Fohler, G., Ramamritham, K. and Fuertes, J.M., *Improving quality-of-control using flexible time constraints: Metric and scheduling issues*. In Proceedings of the 23nd IEEE Real-Time Systems Symposium (RTSS 2002).
10. Yépez, J., Fuertes, J. and Marti, P., *The large error first (LEF) scheduling policy for real-time control systems*. In Work in Progress Proceedings of the 24th IEEE Real-Time Systems Symposium (RTSS WIP 2003), pp. 63-66, (2003).
11. Velasco, M., Fuertes, J. and Marti, P., (2003). *The self triggered task model for real-time control systems*. In Work in Progress Proceedings of the 24th IEEE Real-Time Systems Symposium (RTSS WIP 2003), pp. 67-70, (2003).
12. Lin, C., Marti, P., Brandt, S.A., Banachowski, S., Velasco, M. and Fuertes, J.M., *Improving control performance using adaptive quality of service in a real-time system.* The IEEE Real-Time and Embedded Technology and Applications Symposium (UC Santa Cruz Technical Report UCSC-CRL-04-04), Toronto, Canada, (2004).
13. T. S. Letia, A. Astilean, R. Miron, M.M. Santa, *Train Traffic Control Based on Distributed Resource Allocation*, The Second IFAC Symposium on Telematics Applications, Timişoara, Romania, (2010).
14. Ho, T.K., Ferreira, L. and Law, K.H., *Agent applications in rail transportation*. International Conference on Intelligent Agents Web Technologies and Internet Commerce, pp.251-260, Vienna, (2004).

Leția, T., Hulea, M., Miron, R., *Distributed Scheduling for Real-Time Railway Traffic Control*. IMCSIT 2008, Real-Time Workshop RTS’08, Wisla, ISSN 1896-7094, Polonia, 2008. pp. 679-685, (2008).

IMPLEMENTAREA GESTIONĂRII CĂILOR FERATE CU AJUTORUL SISTEMELOR MULTIAGENT

ABSTRACT: Problema gestionării traficului pe calea ferată este foarte complexă cu implicații majore în societatea modernă. Trenurile sunt corespunzătoare pentru transportul de persoane şi bunuri cu un raport bun între preț şi rapiditate. Creşterea eficienței şi flexibilității căilor ferate necesită noi metode de gestiune. Noile trenuri charter au fost adăugate continuu fără a disturba traseele trenurilor periodice sau să afecteze siguranța. Un sistem de cale ferată este alcătuit dintr-o rețea şine de cale ferată, stații, dispozitive de siguranță (semafoare, senzori de prezență a trenurilor, etc.) şi setul de trenuri care se deplasează. Fiecare tren are ataşat un agent de tren. Aplicațiile distribuite sunt aplicabile în cazul sistemelor mari asemănătoare cu calea ferată. In lucrarea de față este prezentat un model pentru gestiunea rețelei de cale ferată care evoluează într-un mediu dinamic. Lucrarea prezintă de asemenea un algoritm de gestiune distribuită bazat pe negocierea între agenții de tren. Atributele dinamice ale algoritmului sunt cele măsurate şi evaluate în timpul mişcării trenurilor.

Santa Maria-Magdalena, Technical University of Cluj-Napoca, Maria.SANTA@aut.utcluj.ro
Cuibus Octavian’ Technical University of Cluj-Napoca, Octavian.CUIBUS@aut.utcluj.ro
Leția Tiberiu , Technical University of Cluj-Napoca, Romania, Tiberiu.LETIA@aut.utcluj.ro
Enache Mirela , Technical University of Cluj-Napoca, Romania, mirela.enache@yahoo.com