Human Resource Influence for Vehicle Fleet Management in Discrete Transport Systems

Jacek Mazurkiewicz[#1], Tomasz Walkowiak[#2]

[#]*Institute of Computer Engineering, Control and Robotics, Wroclaw University of Technology*
ul. Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, POLAND

[1]Jacek.Mazurkiewicz@pwr.wroc.pl

[2]Tomasz.Walkowiak@pwr.wroc.pl

***Abstract* — The paper describes an approach to analysis of management algorithms in discrete transport systems (*DTS*). The unified containers transported by trucks with the set of time-type assumptions are the essence of the system discussed. The proposed method is based on modelling and simulating of the system behaviour. The income of containers is modelled by a stochastic process. Each container has a source and destination address. The central node is the destination address for all containers generated in the ordinary nodes. Where containers addressed to any border nodes are generated in the central node. We proposed the detailed formal model of the system including a driver as a new member of the system description. All elements are characterized by the proper set of parameters available in the set of states. We present three possible management approaches and we choose one of them to calculate the *DTS* example based on real data.**

***Keywords* — reliability, discrete transport system, Monte-Carlo simulation, human resource, management methods**

I. Introduction

Administration of a large transport system is not a trivial task. The transport systems are characterized by a very complex structure. The performance of the system can be impaired by various types of faults related to the transport vehicles, communication infrastructure or even by traffic congestion [1]. It is hard for human (administrator, owner) to understand the system behaviour. To overcome this problem we propose a functional approach. The transport system is analysed from the functional point of view, focusing on business service realized by a system [16]. The analysis is following a classical [16]: modelling and simulation approach. It allows to calculate different system measures which could be a base for decisions related to administration of the transport systems. The metric are calculated using Monte Carlo techniques [4]. No restriction on the system structure and on a kind of distribution is the main advantage of the method. The proposed model allows to forget about the classical reliability analysis based on Markov or Semi-Markov processes [4] - idealised and hard for reconciliation with practice.

The paper presents an analysis of transport system of the Polish Post regional centre of mail distribution (described in section 2). Base on which we have developed the discrete transport system model presented in section 3. The main service given by the post system is the delivery of mails. From the client point of view the quality of the system could be measured by the time of transporting the mail from the source to destination.

A driver is a new element of the system description. The pointed the set of states to characterise the actual driver position including formal – law-origin aspects: number of hours he or she can work daily for example.

We offer three approaches to system management: based on time-tables, heuristic and focused on soft-computing (section 3). In our opinion the heuristic one seems to be the most adequate to the level of detail discussed in the described work.

The quality of the analysed system is measured by the availability defined as an ability to realize the transportation task at a required time (described in section 4).

The post system is very hard to be analysed by a formal model since it does not lay in the Markov process framework. Therefore, we have used a computer simulation [4] described in section 5. Next (section 6), we give an example of using presented model and simulator for the analysis of the Polish Post regional centre in Wroclaw transport system and discussed the performance of developed simulator.

II. Polish Post Transport System

The analysed transport system is a simplified case of the Polish Post. The business service provided the Polish Post is the delivery of mails. The system consists of a set of nodes placed in different geographical locations. Two kinds of nodes could be distinguished: central nodes (*CN*) and ordinary nodes (*ON*). There are bidirectional routes between nodes. Mails are distributed among ordinary nodes by trucks, whereas between central nodes by trucks, railway or by plain. The mail distribution could be understood by tracing the delivery of some mail from point *A* to point *B*. At first the mail is transported to the nearest to A ordinary node. Different mails are collected in ordinary nodes, packed in larger units called containers and then transported by trucks scheduled according to some time-table to the nearest central node. In central node containers are repacked and delivered to appropriate (according to delivery address of each mail) central node. In the Polish Post there are 14 central nodes and more than 300 ordinary nodes. There are more than one million mails going through one central node within 24 hours. It gives a very large system to be modelled and simulated. Therefore, we have decided to model only a part of the Polish Post transport system – one central node with a set of ordinary nodes.

Essential in any system modelling and simulation is to define the level of details of modelled system. Increasing the details causes the simulation becoming useless due to the computational complexity and a large number of required parameter values to be given. On the other hand a high level of modelling could not allow to record required data for system measure calculation. Therefore, the crucial think in the definition of the system level details is to know what kind of measures will be calculated by the simulator. Since the business service given by the post system is the delivery of mails on time. Therefore, we have to calculate the time of transporting mails by the system. Since the number of mails presented in the modelled system is very large and all mails are transported in larger amounts containers, we have decided to use containers as the smallest observable element of the system. Therefore, the main observable value calculated by the simulator will be the time of container transporting from the source to the destination node.

The income of mails to the system, or rather containers of mails as it was discussed above, is modelled by a stochastic process. Each container has a source and destination address. The central node is the destination address for all containers generated in the ordinary nodes. Where containers addressed to any ordinary nodes are generated in the central node. The generation of containers is described by some random process. In case of central node, there are separate processes for each ordinary node. Whereas, for ordinary nodes there is one process, since commodities are transported from ordinary nodes to the central node or in the opposite direction.

The containers are transported by vehicles. Each vehicle has a given capacity – maximum number of containers it can haul. Central node is a base place for all vehicles. They start from the central node and the central node is the destination of their travel. The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity. Vehicles operate according to the time-table. The time-table consists of a set of routes (sequence of nodes starting and ending in the central node, times of leaving each node in the route and the recommended size of a vehicle). The number of used vehicle and the capacity of vehicles does not depend on temporary situation described by number of transportation tasks or by the task amount for example. It means that it is possible to realize the route by completely empty vehicle or the vehicle cannot load the available amount of commodity (the vehicle is to small). Time-table is a fixed element of the system in observable time horizon, but it is possible to use different time-tables for different seasons or months of the year.

Summarising the movement of the containers in the system, a container is generated with destination address in some of node (source) at some random time. Next, the container waits in the node for a vehicle to be transported to the destination node. Each day a given time-table is realized, it means that at a time given by the time table a vehicle, selected from vehicles available in the central node, starts from central node and is loaded with containers addressed to each ordinary nodes included in a given route. This is done in a proportional way. When a vehicle approaches the ordinary node it is waiting in an input queue if there is any other vehicle being loaded/unload at the same time. There is only one handling point in each ordinary node. The time of loading/unloading vehicle is described by a random distribution. The containers addressed to given node are unloaded and empty space in the vehicle is filled by containers addressed to a central node. Next, the vehicle waits till the time of leaving the node (set in the time-table) is left and starts its journey to the next node. The operation is repeated in each node on the route and finally the vehicle is approaching the central node when it is fully unloaded and after it is available for the next route. The process of vehicle operation could be stopped at any moment due to a failure (described by a random process). After the failure, the vehicle waits for a maintenance crew (if there are no available due to repairing other vehicles), is being repaired (random time) and after it continues its journey. The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity.

III. Discrete Transport System Formal Model

A. Model Overview

The described in the previous section regional part of the Polish Post transport system with one central node and several ordinary nodes was a base for a definition of a formal model of a discrete transport system (*DTS*).

Generally speaking users of the transport system are generating tasks which are being realized by the system. The task to be realized requires some services presented in the system. A realization of the system service needs a defined set of technical resources. Moreover, the operating of vehicles transporting mails between system nodes is done according to some rules – some management system. Therefore, we can model discrete transport system as a 5-tuple:

$$DTS = \langle Client, Driver, BS, TI, MS \rangle \tag{1}$$

Client – client model,
Driver – driver model,
BS – business service,
TI – technical infrastructure,
MS – management system.

B. Technical Infrastructure

During modelling of technical infrastructure we have to take into consideration functional and reliability aspects of the post transport system. Therefore, the technical infrastructure of *DTS* could be described by three elements:

$$TI = \langle No, V, MM \rangle, \tag{2}$$

No – set of nodes,
V – set of vehicles,
MM – maintenance model.

Set of nodes (*No*) consists of single central node (*CN*), a given number of ordinary nodes (ON_i). The distance between each two nodes is defined by the function:

$$distance\colon No \times No \to R_{+}. \tag{3}$$

Each node has one functional parameter the mean (modelled by normal distribution) time of loading a vehicle:

$$loading: No \to R_+ . \quad (4)$$

Moreover, the central node (*CN*) has additional functional parameter: number of service points (in each ordinary node there is only one service point):

$$servicepoints: CN \to N_+ . \quad (5)$$

Each vehicle is described by following functional and reliability parameters:

- mean speed of a journey
  $$meanspeed: V \to R_+ , \quad (6)$$
- capacity – number of containers which can be loaded
  $$capacity : V \to R_+ , \quad (7)$$
- mean time to failure
  $$MTTF : V \to R_+ , \quad (8)$$
  a time when failure occurs is given by exponential distribution with mean equal to a value of *MTTF* function,
- mean repair time
  $$MRT : V \to R_+ . \quad (9)$$

The traffic is modelled by a random value of vehicle speed and therefore the time of vehicle (*v*) going from one node (n_1) to the other (n_2) is given by a formula (10)

$$time(v, n_1, n_2) = \frac{distance(n_1, n_2)}{Normal(meanspeed(v), 0.1 \cdot meanspeed(v))}$$

where *Normal* denotes a random value with the Gaussian distribution.

Maintains model (*MM)* consists of a set of maintenance crews which are identical and unrecognized. The crews are not combined to any node, are not combined to any route, they operate in the whole system and are described only by the number of them. The time when a vehicle is repaired is equal to the time of waiting for a free maintains crew (if all crews involved into maintenance procedures) and the time of a vehicle repair which is a random value with the Gaussian distribution ($Normal(MRT(v), 0.1 \cdot MRT(v))$.

C. Business Service

Business service (*BS*) is a set of services based on business logic, that can be loaded and repeatedly used for concrete business handling process. Business service can be seen as a set of service components and tasks, that are used to provide service in accordance with business logic for this process. Therefore, *BS* is modelled a set of business service components (*sc*):

$$BS = \{sc_1, ..., sc_n\}, n = length(BS) > 0 , \quad (11)$$

the function *length*(*X*) denotes the size of any set or any sequence *X*. Each service component in DTS consist of a task of delivering a container from a source node to the destination one.

D. Client Model

The service realised by the clients of the transport system are sending mails from a source node to a destination one. Client model consist of a set of clients (*C*).

Each client is allocated in one of nodes of the transport system:

$$allocation: C \to No . \quad (12)$$

A client allocated in an ordinary node is generating containers (since, we have decided to monitor containers not separate mails during simulation) according to the Poisson process with destination address set to ordinary nodes. In the central node, there is a set of clients, one for each ordinary node. Each client generates containers by a separate Poisson process and is described by intensity of container generation:

$$intensity: C \to R_+ . \quad (13)$$

The central node is the destination address for all containers generated in ordinary nodes.

E. Driver Model

Each driver is describing by actual state of him/her (s_d)

- rest (not at work),
- unavailable (illness, vacation, etc.),
- available (at work – ready to start driving),
- break (during driving),
- driving.

The number of driver working hours is limited by the labour law. The regulation is rather complicated – depends on a lot of parameters and the number of drivers fixed to the truck for example, but for transport system we analyse we can say that the daily limit for each driver equals to 8 or 9 hours and single driver operates with one truck. This assumption is legal in European Union.

So the problem of working hours (w_h) we can solve as follow:

- if w_h > *limit* then s_d = rest & w_h = 0,
- where *limit* = 8 hours or *limit* = 9 hours,
- *limit* – is the time period of a single change in work.

The single shift can be distinguish as: morning or afternoon. So at 6am for each driver:

- if *shift* == morning & s_d == rest then s_d = available,

So at 1pm for each driver:

- if *shift* == afternoon & s_d == rest then s_d = available,

The next problem ought to be modelled is the driver's illness state. We propose the following approach:

- for every driver at 4am: if s_d == rest,
- if *rand()* < d_i then during *x* days (according to the given distribution) the driver is in s_d = unavailable,
- where d_i – driver's illness parameter.

Moreover we propose to categorise the driver's illnesses as follow:

- short sick: 1 to 3 days,
- typical illness: 7 to 10 days,
- long-term illness: 10 to 300 days.

We record the daily story of the driver. The algorithm to fix the driver to the vehicle is the last part of the driver model:

- if no driver – the vehicle does not start,
- driver can be chosen if: s_d = available and w_h + *estimated time of journey* < *limit* * 1.1,
- the driver is chosen randomly or by least value approach: *abs*(limit – w_h - *estimated time of journey*).

F. Legacy Management System

The management system (*MS*) of the *DTS* controls the operation of vehicle. It consists of a sequence of routes:

$$MS = \langle r_1, r_2, ..., r_{nr} \rangle \, . \tag{14}$$

Each route is a sequence of nodes starting and ending in the central node, times of leaving each node in the route (t_i) and the recommended size of a vehicle (*size*) :

$$\begin{gathered} r = \langle CN, t_0, n_1, t_1, ..., n_m, t_m, CN, size \rangle \\ v_i \in No - \{CN\} \quad 0 \le t_0 < t_1 < ... < t_m < 24h \end{gathered} \tag{15}$$

The routes are defined for one day and are repeated each day. The management system selects vehicles to realise each route in random way, first of all vehicles (among vehicles available in central node) with capacity equal to recommended size are taken into consideration. If there is no such vehicle, vehicles with larger capacity are taken into consideration. If still there is no vehicle fulfilling requirements vehicle of smaller size is randomly selected. If there is no available vehicle a given route is not realised.

The pros and cons of legacy approach to management problem was discussed in our previous papers. The results are available in [19]. In our opinion the solution does not fit to the level of detail if we have the driver as a part of the system.

G. Heuristic Management System

As it was mentioned in the introduction, we proposed a replacement of legacy management system by a heuristic decision algorithm [19]. The decisions (send a truck to a given destination node) are taken in moments when a container arrives to the central node. The truck is send to a trip if:

- the number of containers waiting in for delivery in the central node of the same destination address as that just arrived is larger than a given number,
- there is at least one available vehicle,
- the simulated time is between 6 am and 22 pm minus the average time of going to and returning from the destination node.

The truck is send to a node defined by destination address of just arrived container. If there is more than one vehicle available in the central node, the vehicle with size that a fits the best to the number of available containers is selected, i.e. the largest vehicle that could be fully loaded. If there are several trucks with the same capacity available the selection is done randomly.

On the other hand we observe in the same way the vehicles available in the ordinary nodes. The only difference is the greater level of threshold to initialise the vehicle journey.

The restriction for the time of truck scheduling (the last point in the above algorithm) are set to model the fact that drivers are working on two 8 hours shifts.

H. Soft Computing Management System

As it was mentioned in the introduction we also proposed the other replacement of the legacy management system based on a neural network based [21]. The system consists of a multilayer perceptron to decide if and where to send trucks. The input to the neural network consists of:

$$in = \langle pkc_1, pkc_2, ..., pkc_{npk}, cnc_1, cnc_2, ..., cnc_{npk}, nfv \rangle \tag{16}$$

npk – number of ordinary nodes,

pkc_i – number of containers waiting for delivery in the central node with destination address set to *i*-th ordinary node,

nfv – number of free vehicles in the central node,

Each output of the network corresponds to each ordinary node:

$$nnout = \langle out_1, out_2, ..., out_{npk} \rangle \, , \tag{17}$$

The output of the network is interpreted as follows (for sigmoid function used in output layer):

$$j = \arg\max_{i=1...npk} \{out_i\} \, . \tag{18}$$

If out_j is greater than 0.5 send a vehicle to node *j* else do nothing. If there are more vehicles available in the central node, the largest vehicle that could be fully loaded is selected. If there are available several trucks with the same capacity selection is done randomly. The neural network decision (send a truck or not and where the truck should be sent) are taken in given moments in time. These moments are defined by following states of the system:

- the vehicle comes back to the central node and is ready for the next trip,
- if in central node there is at least one available vehicle and the number of containers of the same destination address is larger than the size of the smallest available vehicle.

The neural network used in the management system requires a learning process that will set up the values of its weights. The most typical learning in the case of multilayer perceptron is the back propagation algorithm. However, it cannot be used here since it is impossible to state what should be the proper output values of the neural network. Since it is hard to reconcile what are the results of a single decision made by the management system. Important are results of the set of decisions. Since the business service realised by transport system is to move commodities without delays, the neural network should take such decisions that allows to reduce delays as much as possible. To train neural network to perform such task we propose to use genetic algorithm [18, 21]. Similar approach to training neural network is applied in case of computer games. The most important in case of genetic algorithm is a definition of the fitness function. To follow business service requirements of transport system we propose following definition of the fitness function calculated for a given neural network after some time (*T*) (therefore after a set of decisions taken by neural network):

$$fitness(T) = \frac{N_{ontime}(0,T) + N_{ontimeinsystem}(T)}{N_{delivered}(0,T) + N_{insystem}(T)}. \qquad (19)$$

It is a ratio of on-time containers (delivered with 24h and being in the system but not longer then 24h) to all containers (that already delivered $N_{delivered}(0,T)$ and still being presented in the system $N_{insystem}(T)$).

The solution described above is very complicated and the first necessary step – learning phase – takes a lot of time and requires a lot of data to create the proper weights and other initial parameters. This is the reason why – taking into account quite good results of the soft computing approach to management [21] – we does not decide to use it the contemporary work.

IV. Functional Metrics of DTS

A. Overview

The formal model described previously was designed to allow to develop a simulator which allows to observe the time of transporting each container. Based on these observation several metrics could be defined. As it was mentioned in the introduction we focus here on the service oriented approach [17]. Therefore we propose that the availability to be a key parameter for the evaluation of the quality of the *DTS*. One can define the availability in different ways, but always the value of an availability can be easy transformed into economic or functional parameters perfectly understood by owner of the system. The availability in mostly understood as a probability that a system is up. And is defined as a ratio of the expected value of the uptime of a system to the observation time. It is simple definition but requires defining what does it mean that transport system is working. The similar metric is the acceptance ratio defined in information since as a number of accepted requests to the total number of requests.

B. Acceptance Ratio

Let introduce the following notation:

- T – a time measured from the moment when the container was introduced to the system to the moment when the container was transferred to the destination (random value),
- T_g – a guaranteed time of delivery, if exceeded the container is delayed.

In [16] we have proposed performance metric – acceptance ratio. It is defined as a ratio of on-time containers (containers for which $T < T_g$) to all containers within a given time of observation (0, τ). Within the time period a given number of containers are delivered ($N_{delivered}(\tau)$), a part of them or all delivered on time ($N_{ontime}(\tau)$), but at the end of analysed period time there could be some containers not yet delivered (waiting in the source node or being transported) ($N_{insystem}(\tau)$) and all or part of them being not late yet ($N_{ontimeinsystem}(\tau)$). Taking into consideration introduced symbols the availability could be calculated as the expected value (Monte-Carlo approach) of ratio of on-time containers to all containers:

$$AR_{\tau} = E\left(\frac{N_{ontime}(\tau) + N_{ontimeinsystem}(\tau)}{N_{delivered}(\tau) + N_{insystem}(\tau)}\right). \qquad (20)$$

V. Discrete Transport System Simulation

A. Event-Driven Simulation

Discrete transport system described in the section 3 is very hard to be analysed by formal methods. It does not lay in the Markov process framework [2]. A common way of analysing that kind of systems is a computer simulation. To analyse the system we must first build a simulation model, which was done based on the formal model presented in the previous section, and then operate the model. The system model needed for simulation, has to encourage the system elements behaviour and interaction between elements.

Once a model has been developed, it is executed on a computer. It is done by a computer program which steps through time. One way of doing it is so called event-driven simulation. Which is based on a idea of event, which could is described by time of event occurring and type of an event. The simulation is done by analysing a queue of event (sorted by time of event occurring) while updating the states of system elements according to rules related to a proper type of event. Due to a presence of randomness in the *DTS* model the analysis of it has to be done based on Monte-Carlo approach [4]. What requires a large number of repeated simulation.

Summarising, the event-driven simulator repeats *N*-times the following loop:

- beginning state of a *DTS* initialization,
- event state initialisation, set time $t = 0$,
- repeat until $t < \tau$:
- take first event from event list,
- set time equals time of event,
- realize the event – change state of the *DTS* according to rules related to proper type of event: change objects attributes describing system state, generate new events and put them into event list, write data into output file.

B. Events and Elements of DTS Simulator

In case of *DTS* following events (mainly connected with vehicles) have been defined:

- vehicle failure,
- vehicle starts repair,
- vehicle repaired,
- vehicle reached the node,
- vehicle starts from the node,
- vehicle is ready for the next route,
- time-table (starting the route in the central node).

The processing of events of done in objects representing *DTS* elements. The objects are working in parallel. Following types of system elements were distinguished: vehicle, ordinary node, central node, time table.

The life cycle of each object consists of waiting for an event directed to this object and then execution of tasks required to perform the event. These tasks includes the changes of internal state of the object (for example when vehicle approaches the node it is unloaded, i.e. the number of hauled containers decreases) and sometimes creating a new even (for example the event vehicle starts from the node generates new event vehicle reached the node – next node in the trip). The random number generator is used to deal with random events, i.e. failures. It is worth to notice that the current analysed event not only generates a new event but also could change time of some future events (i.e. time of approaching the node is changed when failure happens before). The time of a new event is defined by the sum of current time (moment of execution of the current event) and the duration of a given task (for example vehicle repair). Only times of starting a given route (event vehicle starts from the central node) are predefined (according to the time table). Duration of all other tasks are defined by system elements states:

- time when vehicle waits in the queue for loading/unloading,
- time when vehicle waits in the queue for maintains crew, or are given by random processes:
- time of vehicle going between two nodes,
- time of loading/unloading,
- time to failure,
- repair time.

Moreover each object representing a node have additional process (working in parallel) which are responsible for generating containers. The life cycle of this process is very simple: waiting a random time, generating a container with a given destination address (central node for all ordinary nodes, and each ordinary nodes for process in the central node) and storing a container in the store house (implemented as a queue) of a given node.

C. *DTS Simulator Implementation*

The event-simulation program could be written in a general purpose programming language (like C++), in a fast prototyping environment (like *Matlab*) or a special purpose discrete-event simulation kernel. One of such kernels, is the Scalable Simulation Framework (*SSF*) [16] which is a used for *SSFNet* [16, 17] computer network simulator. *SSF* is an object-oriented API - a collection of class interfaces with prototype implementations. It is available in C++ and Java. *SSFAPI* defines just five base classes: *Entity, inChannel, outChannel, Process*, and *Event*. The communication between entities and delivery of events is done by channels (channel mappings connects entities).

For the purpose of simulating *DTS* we have used Parallel Real-time Immersive Modelling Environment (*PRIME*) [17] implementation of *SSF* due to a much better documentation then available for the original *SSF*. We have developed a generic class derived from *SSF* Entity which is a base of classes modelling

DTS objects which models the behaviour of presented in section 2 and 3 discrete transport system.

As it was mentioned a presence of randomness in the *DTS* model, the Monte-Carlo approach is used. The original *SSF* was not designed for this purpose so some changes in *SSF* core were done to allow to restart the simulation from time zero several times within one run of simulation programme.

The statistical analysis of the system behaviour requires a very large number of simulation repetition, therefore the time performance of developed simulator is very important.

VI. DTS Simulation Analysis Results

A. *Exemplar DTS*

We propose for the case study analysis an exemplar *DTS* based on Polish Post regional centre in Wroclaw. We have modelled a system consisting of one central node (Wroclaw regional centre) and twenty two other nodes - cities where there are local post distribution points in Dolny Slask Province. The length of roads were set according to real road distances between cities used in the analysed case study. The intensity of generation of containers for all destinations were set to 4.16 per hour in each direction giving in average 4400 containers to be transported each day. The vehicles speed was modelled by Gaussian distribution with 50 km/h of mean value and 5 km/h of standard deviation. The average loading time was equal to 5 minutes. There was single type of vehicle with capacity of 10 containers. The MTF of each vehicle was set to 2000. The average repair time was set to 5h (Gaussian distribution). [18]

B. *Acceptance Ratio Calculation*

We tried to realise the transportation tasks defined above using: 41, 44, 46 and 48 trucks. The number of drivers to operate the vehicle fleet we changed from 75 to 100. As we can see in Fig. 1. 75 drivers are absolutely inefficient for the analysed system – the acceptance ratio is almost zero and it does not depend on the number of used trucks.

The number of vehicles equal to 44, 46, 48 can be noticed as correct to make the *DTS* operative. The acceptance ratio grows up quickly and reach the value of 0.9 – when we can say that system works “safety-far” from the border of inacceptable state. It is interesting that we need (at the level of acceptance ratio equal to 0.9) only 82 drivers to operate with 48 trucks, 85 drivers if we have 46 trucks and 90 drivers for 41 trucks.

If we require the acceptance ration at the level of 1 – 85 drivers are necessary for 48 trucks, 90 drivers for 46 vehicles and 95 drivers for fleet of 44 trucks.

It is easy to notice that 41 trucks are not acceptable size of vehicle fleet to make the example *DTS* operative. There is no chance to substitute the shortage of trucks by the number of drivers. The acceptance ratio cannot reach the value of 0.6 even if we use 200 drivers.

The example shows how we can tune the size of *DTS* if we know the possible tasks definitions. The *DTS* owner’s decisions ought to be taken in multidimensional environment. Our approach to *DTS* modelling and simulation can make the decision easier – we can observe immediately the results of possible solutions. We final decision ought to be the best – because it always generates financial consequences.

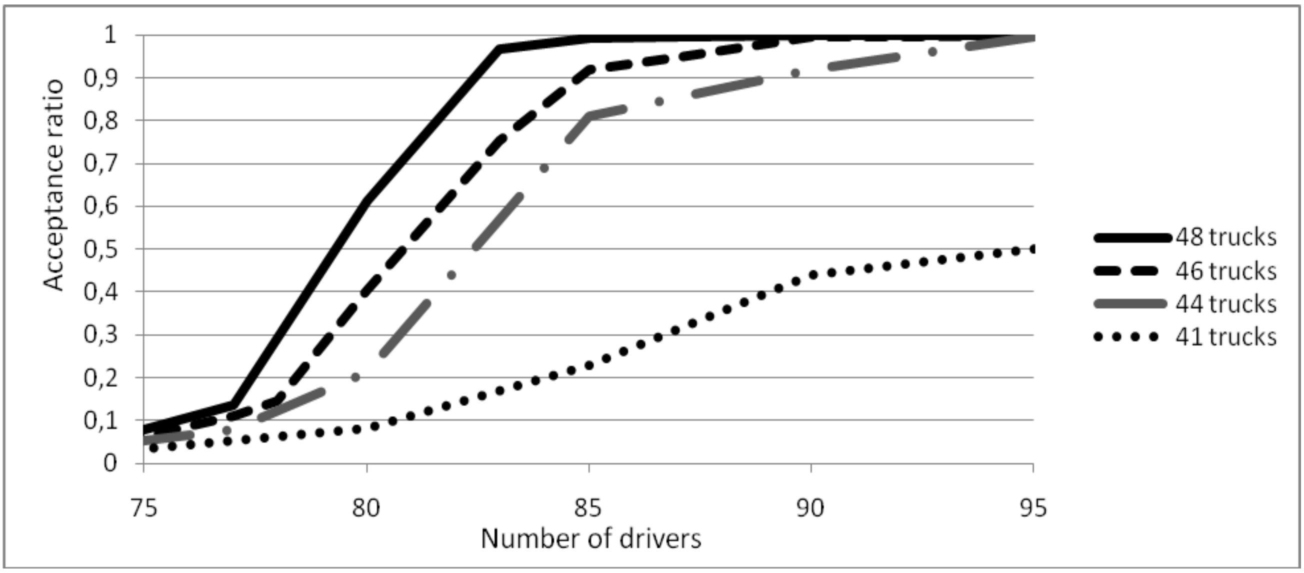

Fig. 1. Acceptance ratio in a function of number of drivers for fixed number of vehicles

VII. Conclusions

We have presented a formal model and event-driven simulator of discrete transport system (*DTS*). The *DTS* model is based on Polish Post regional transport system. Simulator was implemented using the Scalable Simulation Framework (SSF). The simulator allows to perform reliability and functional analysis of the *DTS*, for example:

- determine what will cause a ”local” change in the system,
- make experiments in case of increasing number of containers per day incoming to system,
- identify weak point of the system by comparing few its configuration,
- better understand how the system behaves,
- foresee changes caused by human resource influence.

Based on the results of simulation it is possible to create different metrics to analyse the system in case of reliability, functional and economic case. The availability, average availability and acceptance ratio of the system was introduced - defined in a functional way by delayed tasks realisation.

The metric could be analysed as a function of different essential functional and reliability parameters of *DTS*. Also the system could be analyse in case of some critical situation (like for example a few day tie-up. The paper includes some exemplar systems, based on real Polish Post Wroclaw area, and calculated metric.

The achieved performance of the *DTS* simulator makes it a practical tool for defining an organization of vehicle maintenance and transport system logistics.

Acknowledgment

Work reported in this paper was sponsored by a grant No. N N509 496238, (years: 2010-2013) from the Polish Ministry of Science and Higher Education.

References

[1] Barcelo, J., Codina, E., Casas, J., Ferrer, J.L. and Garcia, D. Microscopic Traffic Simulation: a Tool for the Design, Analysis And Evaluation Of Intelligent Transport Systems, *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 41, 2005, pp. 173-203.

[2] Barlow, R. and Proschan, F . *Mathematical Theory of Reliability*. Philadelphia: Society for Industrial and Applied Mathematics, 1996. 345 p.

[3] Ben-Akiva, M., Cuneo, D., Hasan, M., Jha, M. and Yang, Q. Evaluation of Freeway Control Using a Microscopic Simulation Laboratory, *Transportation Research, Part C (Emerging Technologies)*, Vol. 11C, 2003, pp. 29-50.

[4] Birta, L. and Arbez, G. Modelling and Simulation: Exploring Dynamic System Behaviour. London: Springer, 2007. 246 p.

[5] Burt, C.N. and Caccetta, L. Match Factor for Heterogeneous Truck and Loader Fleets, *International Journal of Mining, Reclamation and Environment*, Vol. 21, 2007, pp. 262-270.

[6] Duinkerken, M.B., Dekker, R., Kurstjens, S.T.G.L., Ottjes, J.A., and Dellaert, N.P. Comparing Transportation Systems for Inter-Terminal Transport at the Maasvlakte Container Terminals, *OR Spectrum*, Vol. 28, 2006, pp. 469-493.

[7] Fishman, G. Monte Carlo: Concepts, Algorithms, and Applications. Springer-Verlag, 1996. 129 p.

[8] Gartner, N., Messer, C.J. and Rathi, A.K. Traffic Flow Theory and Characteristics. In: *T.R. Board (Ed.).* Texas: University of Texas at Austin, 1998. 125 p.

[9] Gold, N., Knight, C., Mohan, A. and Munro, M. Understanding service-oriented software, *IEEE Software*, Vol. 21, 2004, pp. 71-77.

[10] Ioannou, P.A. *Intelligent Freight Transportation*. Carolina: Taylor and Francis Group, 2008. 245 p.

[11] Krzyzanowska, J. The Impact of Mixed Fleet Hauling on Mining Operations at Venetia Mine, *Journal of The South African Institute of Mining and Metallurgy*, Vol. 107, 2007, pp. 215-224.

[12] Liu, H., Chu, L. and Recker, W. Performance Evaluation of ITS Strategies Using Microscopic Simulation. In: *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, 2004, pp. 255-270.

[13] Sanso, B. and Milot, L. Performability of a Congested Urban-Transportation Network when Accident Information is Available, *Transportation Science*, Vol. 33, No 1, 1999, pp. 10-21.

[14] Taylor, M.A.P., Woolley, J.E. and Zito, R. Integration of the Global Positioning System and Geographical Information Systems for Traffic Congestion Studies, *Transportation Research, Part C (Emerging Technologies)*, Vol. 8C, 2000, pp. 257-285.

[15] Vis, I.F.A. Survey of Research in the Design and Control of Automated Guided Vehicle Systems, *European Journal of Operational Research*, Vol. 170, 2006, pp. 677-709.

[16] Walkowiak, T. and Mazurkiewicz, J. Analysis of Critical Situations in Discrete Transport Systems. In : *Proceedings of International Conference on Dependability of Computer Systems*, *Brunow, Poland, June 30-July 2, 2009*. Los Alamitos: IEEE Computer Society Press, 2009, pp. 364-371.

[17] Walkowiak, T. and Mazurkiewicz, J. Availability of Discrete Transport System Simulated by SSF Tool. In: *Proceedings of International Conference on Dependability of Computer Systems, Szklarska Poreba, Poland, June, 2008*. Los Alamitos: IEEE Computer Society Press, 2008, pp. 430-437.

[18] Walkowiak, T. and Mazurkiewicz, J. Functional Availability Analysis of Discrete Transport System Realized by SSF Simulator. In: *Computational Science – ICCS 2008, 8th International Conference, Krakow, Poland, June 2008*. Springer-Verlag, LNCS 5101, 2008. Part I, pp. 671-678.

[19] Walkowiak, T. and Mazurkiewicz, J. Algorithmic Approach to Vehicle Dispatching in Discrete Transport Systems. In: *Technical approach to dependability / Ed. by Jarosław Sugier, et al.* Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej, 2010. pp. 173-188.

[20] Walkowiak, T. and Mazurkiewicz, J. Functional Availability Analysis of Discrete Transport System Simulated by SSF Tool, *International Journal of Critical Computer-Based Systems*, Vol. 1, No 1-3, 2010, pp. 255-266.

[21] Walkowiak, T. and Mazurkiewicz, J. Soft Computing Approach to Discrete Transport System Management. In: *Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2010. Vol. 6114, pp. 675-682.