Transport and Telecommunication, 2011, Volume 12, No 3, 14–26
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia

MULTIAGENT MODELING AND XML-LIKE DESCRIPTION OF DISCRETE TRANSPORT SYSTEM

Katarzyna Nowak, Jacek Mazurkiewicz

Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology
ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
E-mails: Katarzyna.M.Nowak@pwr.wroc.pl, Jacek.Mazurkiewicz@pwr.wroc.pl

The paper describes a novel idea to monitoring and modeling of Discrete Transport System (*DTS*). We propose the formal model of the transport system with the approach to its modelling based on the system behaviour observation. The proposed method is based on specified description languages that can be seen as a bridge between the system description (as a mathematical or expert specification) and the analysis tools. The *DTS* is a simplified case of the Polish Post. Using a multilevel-agent based architecture the realistic data are collected. Described architecture can be found as a basis for a tool that can visualize and analyze data, with respect to the real parameters. Absence of any restrictions on the system structure and on the kind of distribution describing the system functional and reliability parameters is the main advantage of the approach. The paper presents some exemplar system modeling based on a case study. The proposed approach can be used as a helpful tool to administrate the system.

Keywords: Discrete Transport System, reliability, agent management system

1. Introduction

The transportation systems are characterized by a very complex structure. This complexity affects a need to use mechanisms and helpful means to actual data collecting and to administrate the system (i.e. make decisions related to the performance). The building or the optimization of the transportation systems can be expensive and problematic.

The necessary analysis mechanisms should be created not only for the money saving, by also as a tool for the future administration of the system and decision support (based on some specified metrics). The main problem is to realize multi-criteria optimization for the transport system management. The solution ought to combine the sets of reliability, functional and economic parameters. The mentioned data are modeled by distributions, so it makes the optimization problem more sophisticated. The classic solution way is based on the human-expert experience to play the dispatcher or the manager role. But nowadays the elements of the transportation systems are characterized by more and more screwed-up parameters and the historical experience is not enough to create the real and actual solution for the transportation systems driving. This is the reason why we propose the computational collective intelligence to create the device to support human's decisions.

The presented work uses the agents in task of the transportation system monitoring and modelling, so we propose the following description of the most important agent features [13]:

- unique identification within the proposed architecture,
- interaction abilities and proper interfaces for communication and different data transfer,
- secure protocols necessary for communication purposes,
- hardware and / or software implementation,
- plug-and-play ability to guarantee promising scalable and flexible structure.

The temporary computer engineering still does not define an "agent" term in detailed way, but it is not a real barrier to establish the unified semantic meaning of the word in technical point of view. The agent can play the role of the autonomous entity [4] as a model or software component for example. The agent behaviour can be noticed as trivial reactions, but it is not limited – so we can easily find agents characterised by the complex adaptive intelligence. Sometimes it is important to point the potential adaptive abilities of the agents [7]. It means the agent can gather knowledge from the environment around and to adjust its behaviour as a reaction for different events.

This way we can say the agents belong to the soft-computing world. The agent structure is not obligatory plain. We can easily [2] find at least two levels (lower, higher) of the rules created for the agents. This approach allows adjusting the level of the sensitivity for the environment and defining the vitality feature of the agent understood as activity or passivity [9, 10].

The agent-based approach provides the really great effectiveness comparing to the classical architectures if we think about the data gathering and aggregation from the really sophisticated system characterised by the large network, significant number of nodes and non-trivial addressing aspects. This way it is easy to create the global and detailed enough view for multilevel systems with elements described by the various sets of features. We propose to use the agents to create the intelligent hierarchical monitoring architecture (described in Section 3) for the Discrete Transport System (*DTS*) defined in Section 2.

The section 4 presents an authors' solution of a description language for a proposed model, called *DTSML* (Discrete Transport System Modelling Language). The format of the proposed language *XML* (Extensible Markup Language) has been chosen. The main reason is a simple (easy to learn) and readable structure, that can be easily convert to the text or other format.

Moreover, *XML* is supported not only with various tools (providing validation possibilities) but is also supported with many programming languages and framework in case of quicker and more efficient implementation.

Section 5 is focused on simulation methodology. The approach is based on Monte-Carlo technique. The simulator is constructed using *SSF Net* device by adding the necessary objects to reflect the mean features characterizing Discrete Transport Systems.

Next section defines the availability as a metrics to evaluate the quality of the system. The availability is estimated basing on the delays noticed during the system work. Finally we present the case study with a set of results calculated using the approach described in the paper.

2. Transport Model

2.1. Preface

Depending on the level of detail in modelling the granularity of the traffic flow, the traffic models are broadly divided into two categories: macroscopic and microscopic models. According to Gartner et al. [5], a macroscopic model describes the traffic flow as a fluid process with aggregate variables, such as flow and density. The state of the system is then simulated using the analytical relationships between average variables such as a traffic density, a traffic volume, and average speed.

On the other hand, a microscopic model reproduces interaction of punctual elements (vehicles, road segments, intersections, etc) in the traffic network [1]. Each vehicle in the system is emulated according to its individual characteristics (length, speed, acceleration, etc.) [3]. Traffic is then simulated, using processing logic and models describing vehicle driving behaviour, such as car-following and lane-changing models. These models reproduce the driver-driver and driver-road interactions [8].

Despite its great accuracy level, for many years this highly detailed modelling has been considered a computationally intensive approach. For the last twenty years, with the improvements in processing speed, this microscopic approach has become more attractive [12].

2.2. System Description

The analyzed transportation system is a simplified case of the Polish Post. The business service [14], [6], [17] provided the Polish Post is the delivery of mails. The system consists of a set of nodes placed in different geographical locations. We have the headquarter (*HQ*) located in the central part of Poland and two kinds of nodes can be distinguished: the central nodes (*CN*) and the ordinary nodes (*ON*).

There are bidirectional routes between the nodes. Mails are distributed among the ordinary nodes by trucks, whereas between the central nodes by trucks, railway or by plane. The mail distribution can be understood by tracing the delivery of some mail from point *A* to point *B*.

At first the mail is transported to the nearest ordinary node *A*. Different mails are collected in the ordinary nodes, packed in larger units called containers and then transported by trucks scheduled according to the management architecture decision to the nearest central node.

In the central node the containers are repacked and delivered to the appropriate (according to the delivery address of each mail) central node. In the second – the closest to the destination place – central node the mail is again repacked and delivered in a container to the destination ordinary node.

The headquarter collects all data about the actual situation in the whole transportation system and makes the necessary decisions as the reaction for the temporary needs. The headquarter is not in use in transportation action – if we think about the loading, unloading processes, etc. The central nodes aggregate the data from the single region of the country. And finally the ordinary nodes control the local situation to the end user. The scale of necessary actions depends on the actual needs.

In the Polish Post there are 14 central nodes and more than 300 ordinary nodes. There more than one million mails going through one central node within 24 hours. It gives a very large system to be modeled and simulated.

The process of any system modeling requires defining the level of details. Increase of the details of the system causes the simulation to become useless due to the computational complexity and a large number of the required parameter values to be given.

2.3. Model

We can model Discrete Transport System described above as a 4-tuple [15]:

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

where: *Client* – client model, *BS* – business service, a finite set of service components, *TI* – technical infrastructure, *MA* – monitoring architecture.

The technical infrastructure of *DTS* can be described by three elements:

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

where: *No* – the set of nodes; *V* – the set of vehicles; *MM* – the maintenance model.

The set of nodes (*No*) consists of a single headquarter (*HQ*), the central nodes (*CR*), a given number of ordinary nodes (*ON*). The distance between each two nodes is defined by the function:

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

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

$$loading : No \to R_+ . \tag{4}$$

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

$$servicepoints : CN \to N_+ . \tag{5}$$

Each vehicle is described by the following functional and reliability parameters [16]:

- mean speed of a journey

$$meanspeed : V \to R_+ , \tag{6}$$

- capacity – number of containers which can be loaded

$$capacity : V \to R_+ , \tag{7}$$

- mean time to failure

$$MTTF : V \to R_+ , \tag{8}$$

a time when failure occurs is given by exponential distribution with mean equal to a value of *MTTF* function,

- mean repair time

$$MRT : V \to R_+ . \tag{9}$$

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 another (n_2) is given by a formula:

$$time(v, n_1, n_2) = \frac{distance(n_1, n_2)}{Normal(meanspeed(v), 0.1 \cdot meanspeed(v))} , \tag{10}$$

where *Normal* denotes a random value with the Gaussian distribution [17].

The maintenance 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·MRT(v)).*

Business service [17] (*BS*) is a set of services based on a 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 the business logic for this process. Therefore, *BS* is modelled by a set of business service components (*sc*),):

$$BS = \{sc_{1,} ..., sc_n\}, n = length(BS) > 0 , \tag{11}$$

the function *length*(*X*) denotes the size of any set or any sequence .*X*

The service implemented by the clients of the transport system, sending the mails from a source node to a destination one [15]. Client model consist of a set of clients (*C*).

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

$$allocation : C \to No . \tag{12}$$

A client allocated in an ordinary node is generating containers (since we have decided to monitor the containers, but not the separate mails during the 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 the containers by a separate Poisson process and is described by the intensity of container generation:

$$intensity : C \to R_+ . \tag{13}$$

The management model controls the movement of trucks. We proposed a promising and effective heuristic management approach [14] which allows reacting for the critical situations which can occur during the normal system work [15]. The decisions (to send a truck to a given destination node) are taken in the 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 a destination address of just arrived container. If there is more than one vehicle available in the central node, the vehicle with size that fits best to the number of available containers is selected, i.e. the largest vehicle that can 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 initialize the vehicle journey.

3. Monitoring Architecture

In case of Monitoring Architecture representation and distributed multilevel agent-based architecture can be constructed. Figure 1 shows the diversification of complexity of a system into layers and their placement in a system.

The lowest components of the structure are *Node Probes* which are the simplest pieces of the architecture representing resident level (i.e. vehicles). These are the simplest and easy to get data that at this level represent small value that is why they are aggregated in upper units, forwarded to appropriate supervising *Node Sensors*.

Next *Node Sensors* collect the data and create an image of the particular area – so they are located in the ordinary nodes (*ON*). Again the information is sent to a higher level – *Local Agent* – combined to the central nodes (*CN*).

$$NS_i = \bigcup_j NP_j ;\ j \in N. \tag{14}$$

This set of information creates a database building representation of the local part of the system (sub-network). It means that the local view of the system and partial administration in the system can be done at this level.

$$LA_i = \bigcup_j NS_j ;\ j \in N . \tag{15}$$

The highest components of this structure are the *Global Agent* – working in the headquarter (*HQ*), that picks and processes the local information and views one central unit.

$$GA = \bigcup_j LA_j ;\ j \in N . \tag{16}$$

This module stores all information from the whole system. It is situated in one point and one dedicated machine (with a strong backup). Assembling all local views at this level we get one homogenous global view. At this level, data-mining techniques can be used since we know the politics of the Transportation Company as much all information is about parcels.

Figure 2 shows the same tree of hierarchy, where we can see that *Node Probes* can be seen as the lowest point in the system – cars, tracks, etc. Next the nearest or small office can be presented as a *Node Sensors*. Each Post – ordinary node (*ON*) area belongs to a bigger Post Office (mainly located in a big city) – called the central node (*CN*) during our analysis.

Afterwards, all information and some packages (i.e. in case of international packages) are sent to Central Unit – the headquarter (*HQ*) located in one Central point in the Country.

Looking at Figure 1 and Figure 2, we can see that the set of information flow goes to the central unit – *Global Agent*. For this reason it is the most complex and the simplicity of the data that are needed to describe the system in this point is the highest in hierarchy.

4. Transport Language

Since the purpose of the work is to analyze the transportation system based on a specified mathematical model, there is a need to transfer the data into a format that would be useful in an analysis tool. It requires specifying the data format that can be easily shared between various tools or even several transport architectures (independent form complexity). Several data sharing and exchange standards have been developed in the Intelligent Transport Systems [8].

They define a standard data format for the sharing and exchange of the transportation data mostly based on *UML* (Unified Modeling Language) diagrams. Other solutions, i.e. Japanese standard called *UTMS* (Universal Traffic Management Systems) focuses rather on the road traffic system.

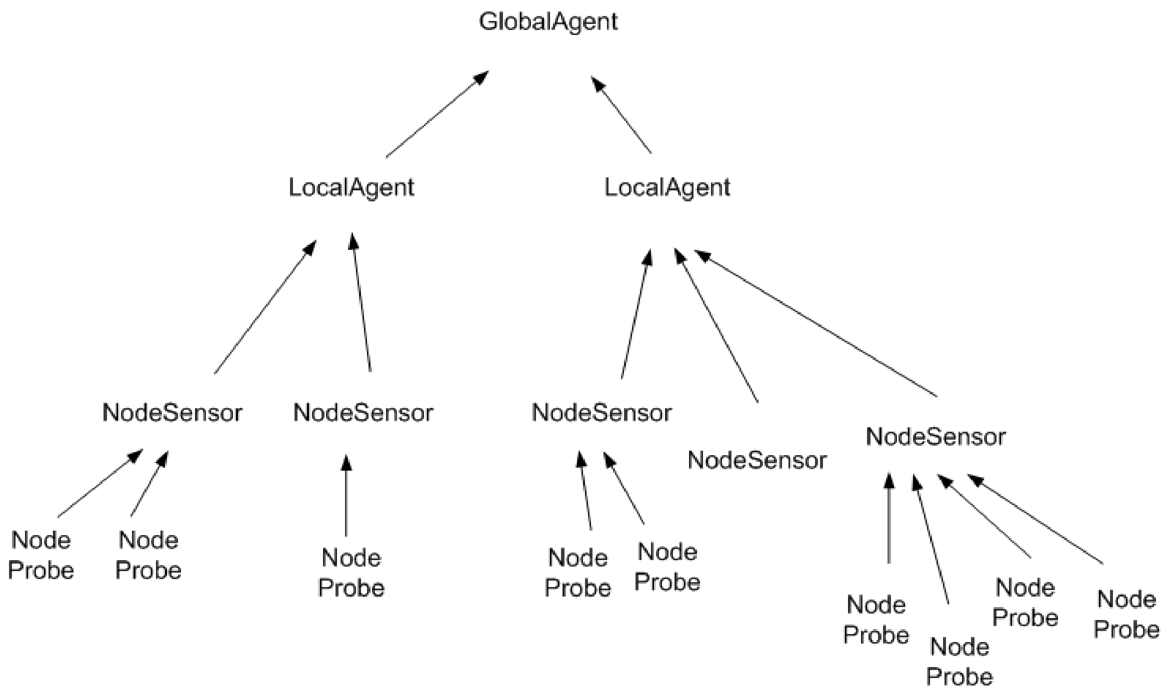

Figure 1. Multilevel architecture scheme

Still none of them is coherent with the solutions proposed in this paper, since they describe the different types of transport system. Moreover, they are based on *UML* diagrams, which are the graphical representation of a model, but not the one, that can be simply used as an input format for any available analysis tool (computer simulator). Additionally description language for this system should be as close as real, not only to a mathematical description of the system, but to the real system behavior and its parameters.

In Section 3 we have mentioned that the View of the Transport System can be implemented on two levels (local and global). To do that, the tool for visualisation and data processing is needed. Furthermore, having this tool we can not only see the topology of the system, but also its elements and parameters.

It gives us an opportunity to see the system more precisely or even make some analysis on a real data that comes from the proposed multilevel agent-based architecture.

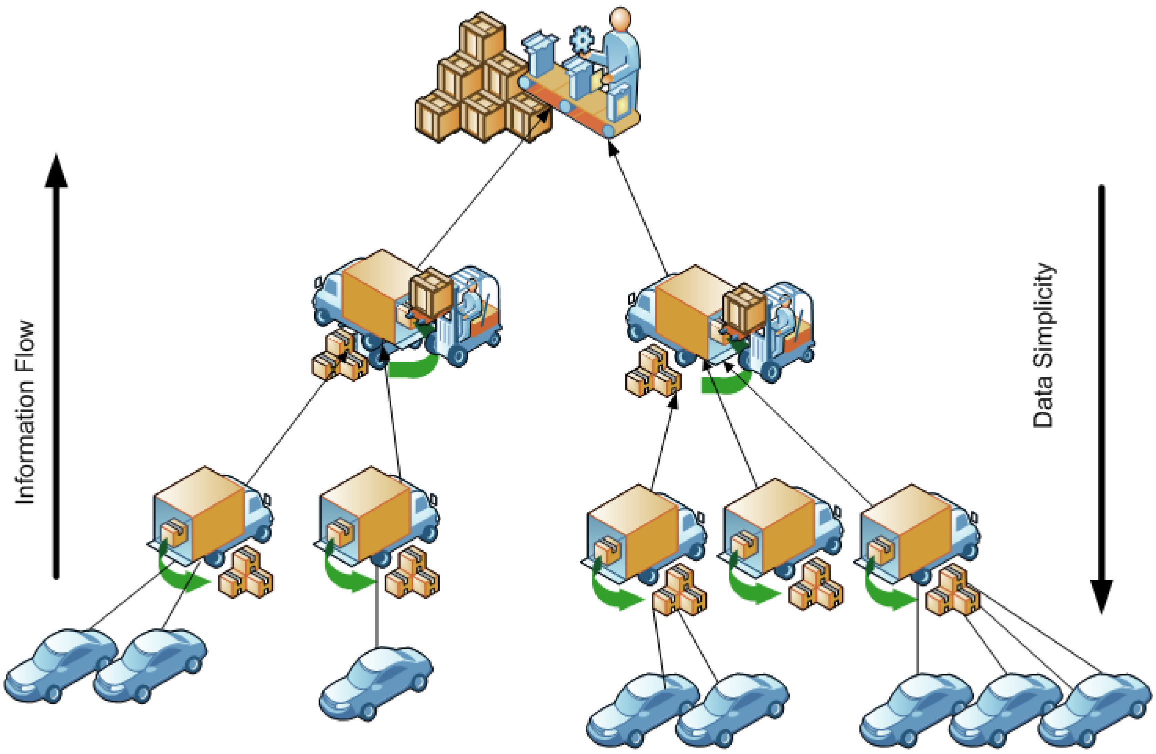

Figure 2. Architecture concept

Still, it requires specifying data format that can be shared between tools, but since of the data exchange is done basing on *UML* diagrams, there is a need to use some other solution that will be more suitable. Since *UML* diagrams are mostly graphical representation of a model, we propose an authors' solution of a *description language* for a proposed (Section 2.3) model, called *DTSML* (Discrete Transport System Modelling Language) [11]. Format of this language is based on *XML* standards, since it is easy to use, and extendable. Moreover, the format allows using the language without special tools, since *XML* is supported by many tools.

Figure 4 shows a fragment of the language with the appropriate elements and attributes related to the mathematical model described in Section 2.3. Figure 3 shows a part of the XML Scheme for Discrete Transport System Modeling Language. As it can be seen, each element of the system described in Section 3 is modeled as a complex element with the appropriate sub-elements and attributes.

The proposed language assures aggregation of dependability and functionality aspects of the examined systems. One language describes the whole system and provides a universal solution for various techniques of computer analysis as an effective and suitable input for those tools.

Expected easiness of potential soft-computing analysis, promising scalability and portability (between analysis tools) can be named as a main advantage of the language usage.

The proposed language is easy to read and to process using popular and open-source tools; however the metadata used in this format are still a significant problem in case of file processing (large size of the file). Nevertheless, since *XML* format is strongly supported by programming languages like: *Java*, *C#*, the usage as much as processing of the file can be done irrespectively to the application language.

As previously described (Fig. 1.) data send by *Node Probe* are combined in the *Node Sensors*. Each of these entities has assigned to it a supervisor – *Local Agent* that accumulates these files in order to create a local view.

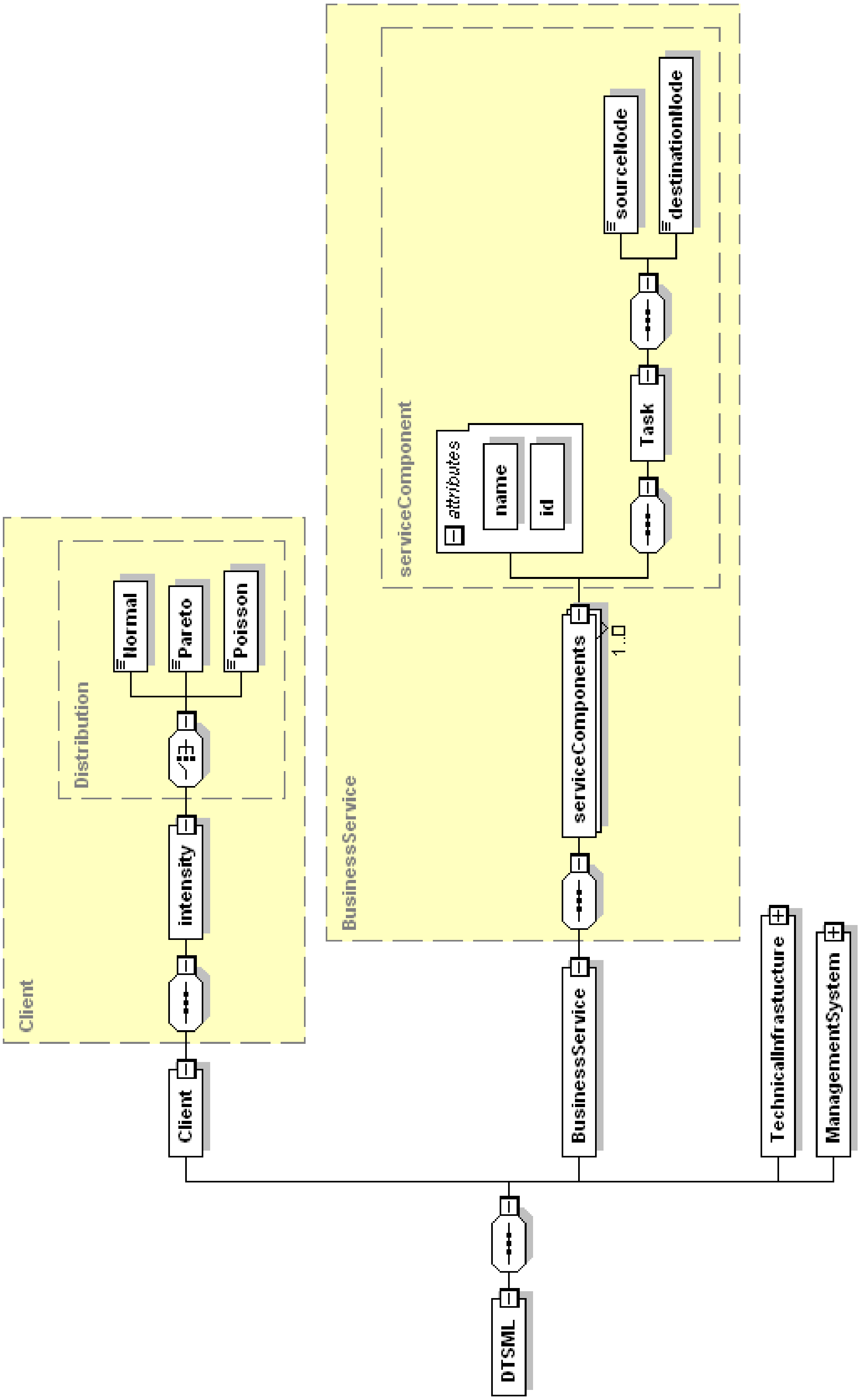

Figure 3. XML Scheme for Discrete Transport System Modeling Language

```
<Node>
         <SingleCentralNode to="string1">
          <numberOfPackages>3455</numberOfPackages>
          <numberOfVehicules>8990</numberOfVehicules>
          <ManagementSystem />
          <TechnicalInfastuctureTopology numberofOrdinaryNodes="5819">
                   <timeBetweenSpecificNodes>
                       <linksBetweenNodes from="Wroclaw" to="Opole" />
                        <time>5.7</time>
                    </timeBetweenSpecificNodes>
                </TechnicalInfastuctureTopology>
            </SingleCentralNode>
        </Node>
        <Vehicule>
            <meanspeed>9.7</meanspeed>
            <capacity>678</capacity>
            <MTTF>10.05</MTTF>
            <MRT>50.98</MRT>
        </Vehicle>
```

Figure 4. *DTSML* – fragment of the language

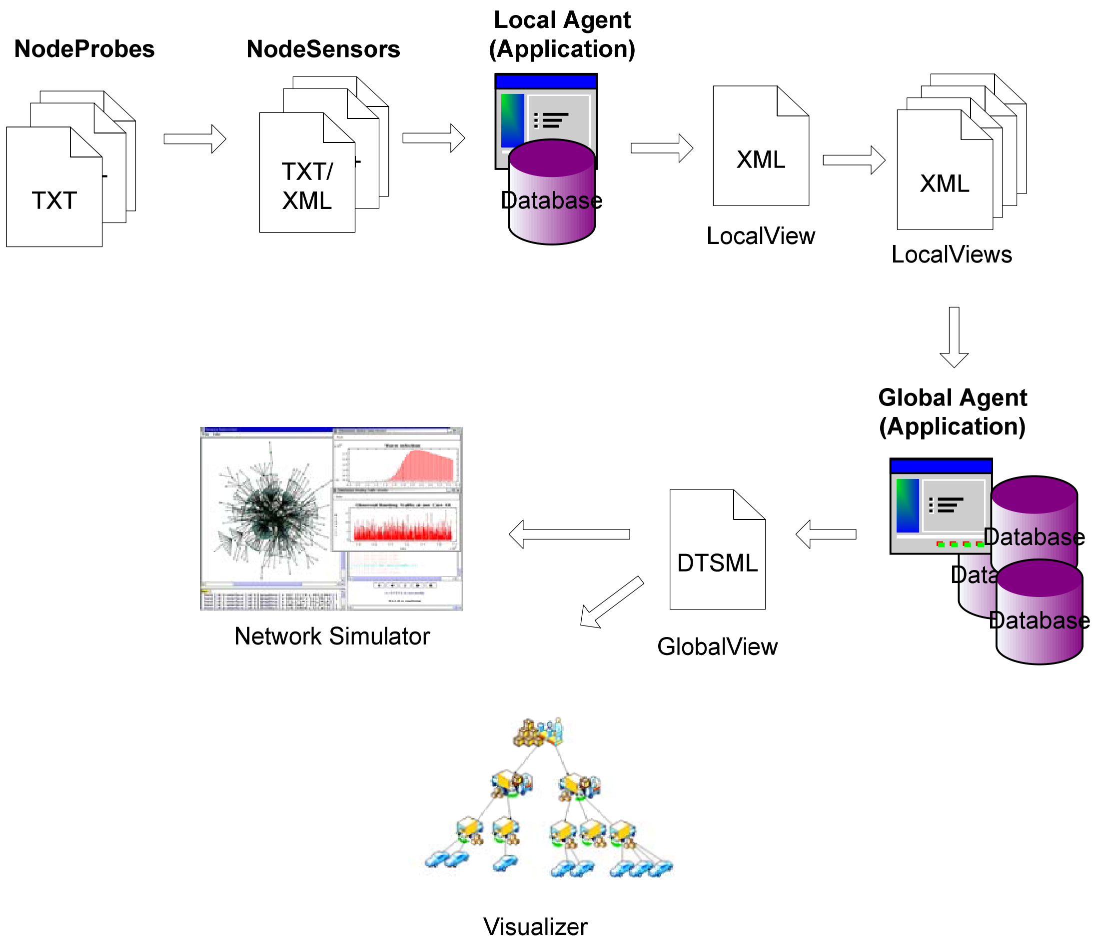

Figure 5. Multi-agent Monitoring Application – architecture concept

This level is more compound and computationally complex than the previous one considering the installed database and some methods that solve additional problems. In this way *XML* files transferred from the simplest level to the next one, creating views on the upper level.

Global Agent collects this information and similarly to Local Agent combines all information included in the dedicated *DTSML* file. As this Global Agent is the most resourceful entity it may be distributed, so it can contain more than one database. At the end, full description of the system is created, visualised and analysed with respect to the dedicated analysis tools.

5. Simulation Methodology

A simulator, performing a Monte-Carlo simulation [16], is composed of four basic elements: input data file, system description language analyzer, event-driven simulator, output data file. The system description language analyzer creates, basing on the data taken from the input file objects which represent system in the memory.

The event-driven simulator repeats *N*-times the following loop [17]:

- initial state of a *DTS* – event initial state, set $t = 0$ – repeat until $t < T$:
- take first event from event list – set time equals time of event – implement the event.

The event is a basis for a simulation process. It is described by the following data: time of event occurring, type of event – vehicle failure for example, part of the *DTS* where event has its influence. The events are placed in the ordered list. Time of event occurring is the key for the list order.

We have the following events in the *DTS*: vehicle reached the node, vehicle is failing, vehicle is repaired, task is starting, end of simulation [14]. For the purpose of simulating *DTS* we have used Parallel Real-time Immersive Modeling Environment (*PRIME*) [16] implementation of *SSF* due to much better documentation then that available for the original *SSF* [14].

We have developed a generic class (named *DTSObject*) derived from *SSF Entity* [6] which is a base of classes modeling *DTS* objects like: scheduler, node, truck and crew which model the behavior of presented in Section 2 Discrete Transport System. Due to a presence of randomness in the *DTS* model the analysis of it has to be done basing on Monte-Carlo approach. It requires a large number of repeated simulations.

The *SSF* is not a Monte-Carlo framework but by simple re-execution of the same code (of course we have to start from different values of random number seed) the statistical analysis of the system behavior can be realized [10]. Data stored in the output file can be used for different measures calculations.

The presented approach to simulation can be enhanced to global hybrid system to analyze the DTS (Fig. 6). We are going to create the modules to combine the economic aspects, dispatching problems and the feedback from the actual situation noticed in the system manoeuvre.

6. System Availability

One can define the availability in different ways, but the value of availability can always be easily transformed into economic or functional parameters perfectly understood by owner of the system.

The availability is mostly understood as a probability that the 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 a simple definition but it requires defining what it means that the transportation system is working.

The similar metric is the acceptance ratio defined by information as a number of accepted requests to the total number of requests.

In paper [16] we have proposed the definition of up time as a time when the number of delayed containers does not exceed a given threshold. Let us introduce the following notation:

- T –a time measured from the moment when the container has been introduced to the system to the moment when the container has been transferred to the destination (random value),
- T_g – a guaranteed time of delivery, if exceeded the container is delayed.
- $N_{delayed}(t)$ – a stochastic process describing the number of delayed containers at time t, i.e. the number of containers for which $T > T_g$.

Therefore, the functional availability $A_k(t)$ can be defined as a probability that the number of delayed containers at time t does not exceed k, the value k is the level of acceptable delay:

$$A_k(t) = \Pr\{N_{delayed}(t) \le k\}. \tag{17}$$

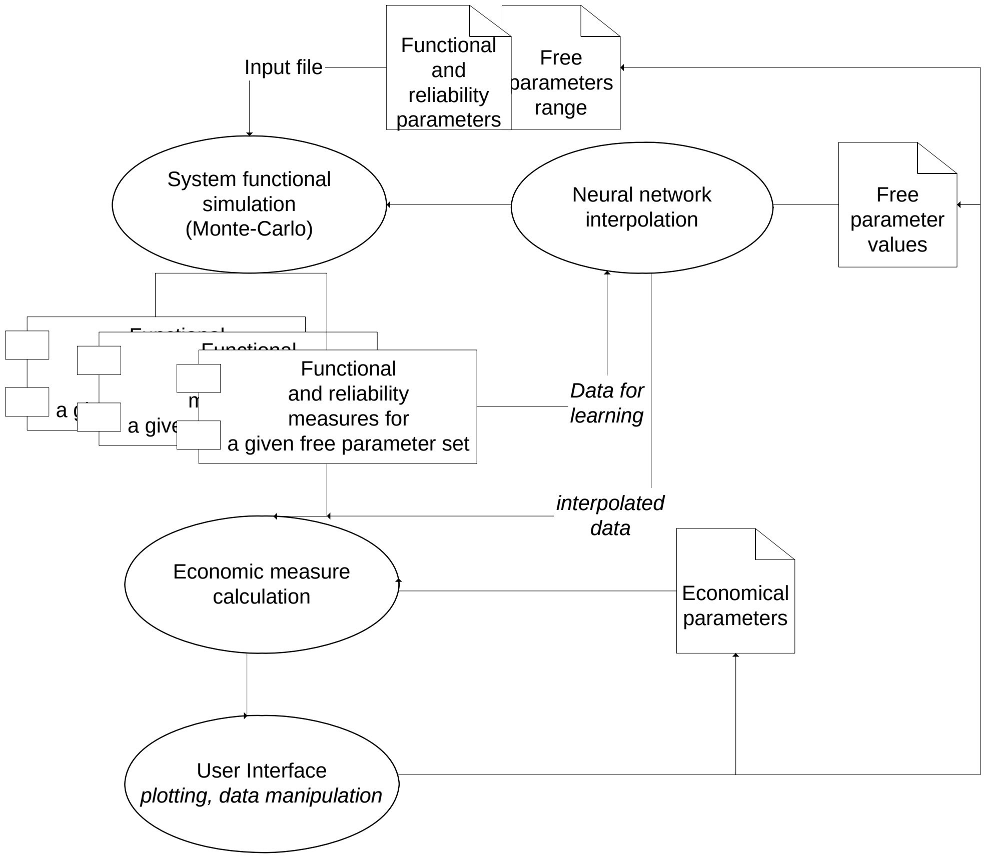

Figure 6. Idea of the global hybrid system for DTS analyzing

7. Case Study

For testing purposes of the presented system (section 2) exemplar transport system has been proposed. It consists of one central node (city Wroclaw, Poland) and three ordinary nodes (cities nearby Wroclaw: Rawicz, Trzebnica and Opole). The distances between the nodes have been set approximating the real distances between used cities and they equal to: 85, 60 and 30 km. We assumed a usage of 5 trucks (two with capacity set to 10 and three with capacity 15) with mean speed of 50km/h.

The vehicles have 19 trips a day: from the central node to the ordinary node and the return trip (i.e. Wroclaw-Opole). Failures of the trucks have been modeled by exponential distribution with mean time of failure equal to 1000h.

The repair time has been modeled by normal distribution with mean value equal to 2h and variance of 0.5h. The containers addressed to the ordinary nodes have been available in the central node at every 0.5, 0.4 and 0.3 of an hour respectively. Containers addressed to the central node have been generated at every 0.6, 0.4, and 0.3 of hour in the following ordinary nodes.

There has been a single maintenance crew. The availability of the system $A_k(t)$ has been calculated with guaranteed time $T_g = 24$h and parameter $k = 20$. Basing on 10 000 time simulations (in each 100 days) the availability of the system has been calculated. The results presented in Fig. 6, Fig. 7 show the periodic changes.

The situation is an effect of used method of containers generation. The containers are generated during all day (by Poisson process) but according to the management system assumptions trucks do not operate at night. The probability of delay increases at night, but the selected number of trucks (5) is satisfactory for the given system. We have also analyzed a system with a reduced number of vehicles (with 4). The resulting value of the availability function is presented also in Fig. 7 and Fig. 8.

It can be noticed that the availability of the system decreases due to the lack of sufficient number of trucks. It should be noticed here that looking at the used management rules and not taking into consideration a randomness of the transport system (failures and traffic jams) only three vehicles can be enough to transport all the generated containers.

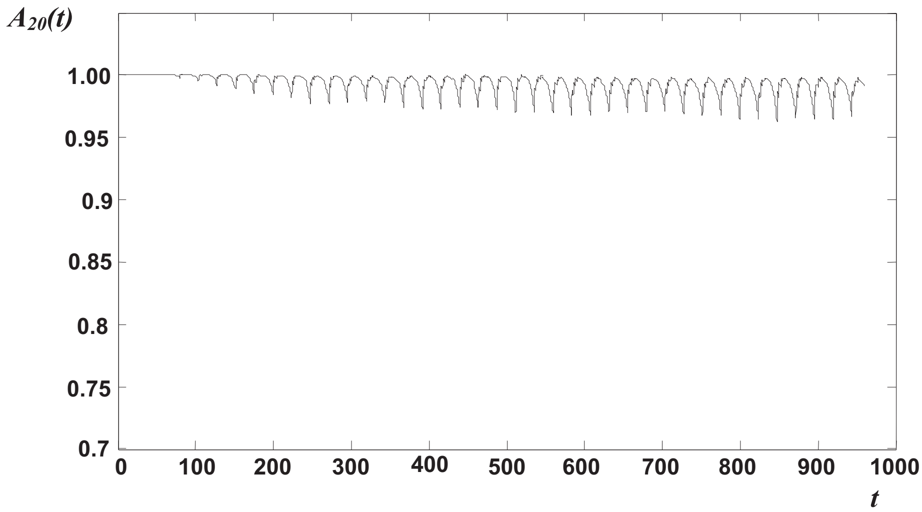

Figure 7. Functional availability of *DTS* – 5 vehicles

8. Conclusion

We have presented a formal model of discrete transportation system (*DTS*) including reliability and functional parameters. The *DTS* model is based on the Polish Post regional transportation system. Using a multilevel-agent based architecture the realistic data are collected and presented in one common language that is an authors' solution.

The proposed description language is helpful for the creation of a computer-aided analysis tools processing according to *DTS* model. The main advantage is the language supporting the further usage of analysis tools.

Moreover, the format of the proposed language is fully expendable since wide support of the *XML* tools. For this reason it can be seen as a base for further (even more precise) system description in case of both dependability and detailed description of the system.

The described architecture can be found as an idea of a tool that can visualize and analyze data, with respect to real parameters. Absence of any restrictions on the system structure and on a kind of distribution describing the system functional and reliability parameters is the main advantage of the approach.

The proposed dependability metrics can be used as an example of an attempt for reaching dependability and functional characteristics as it has been shown in a case study. Further work is to extend the proposed model (DTSML language) with more detailed description that allows even more detailed system specification of its accurate behavior.

The presented solution can be used as a practical tool for defining an organization of vehicle maintenance and transportation system logistics.

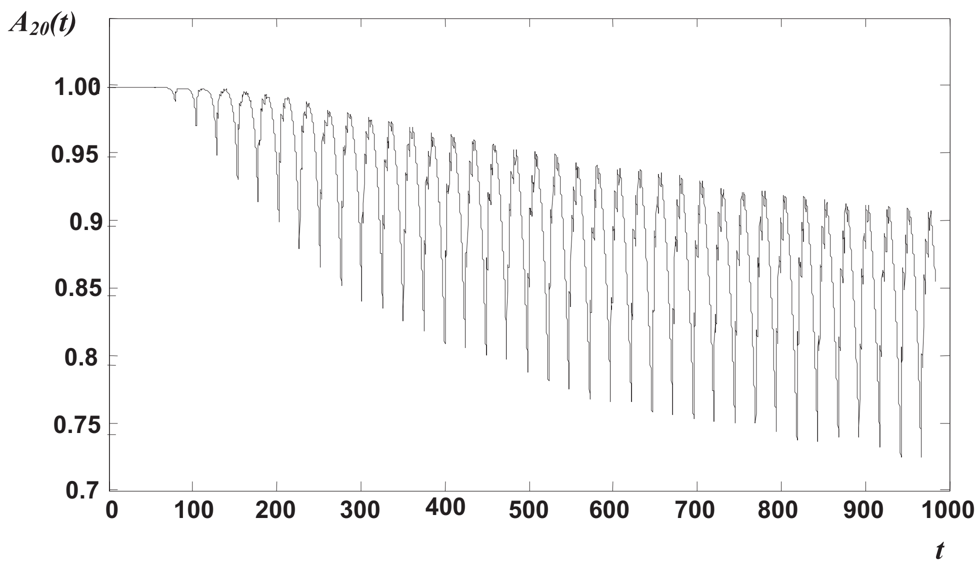

Figure 8. Functional availability of *DTS* – 4 vehicles

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 Transportation Systems, *Journal of Intelligent and Robotic Systems: Theory and Applications*, 41, 2005, pp. 173–203.
2. Bonabeau, E., Agent-based modelling: methods and. techniques for simulating human systems, presented at. *Proc Natl Acad Sci*, 2002.
3. Duinkerken, M. B., Dekker, R., Kurstjens, S. T. G. L., Ottjes, J. A., and Dellaert, N. P. Comparing Transportation Systems for Inter-Terminal Transportation at the Maasvlakte Container Terminals, *OR Spectrum*, 28, 2006, pp. 469–493.
4. Gao, Y., Freeh, V. W., Madey, G. R. Conceptual Framework for Agent-based Modelling and Simulation, *Proceedings of NAACSOS Conference*, Pittsburgh, 2003.
5. Gartner, N., Messer, C. J. and Rathi, A. K., *Traffic Flow Theory and Characteristics*, in T.R. Board, ed., University of Texas at Austin, Texas, 1998, 125 p.
6. Ioannou, P. A. *Intelligent Freight Transportation*, Taylor and Francis Group, Carolina, 2008, 245 p.
7. Jennings, N. R., On Agent-Based Software Engineering, *Artificial Intelligence 117*, Elsevier Press, April 2000, pp. 277–296.
8. Liu, H., Chu, L. and Recker, W., Performance Evaluation of ITS Strategies Using Microscopic Simulation, *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, 2004, pp. 255–270.
9. Mascal, C. M., North, M. J., Tutorial on Agent-Based Modelling and Simulation, *Winter Simulation Conference*, December 2005.
10. Mellouli, S., Moulin, B., Mineau, G. W., Laying Down the Foundations of an Agent Modelling Methodology for Fault-Tolerant Multi-agent Systems, *ESAW 2003*, pp. 275–293.
11. Michalska, K., Mazurkiewicz, J., Functional and Dependability Approach to Transport Services Using Modeling Language; *Computational Collective Intelligence – Technologies and Applications – ICCCI 2011, 3rd International Conference*, Gdynia, Poland, September 2011, Proceedings, Part II, LNAI 6923, Springer-Verlag Berlin Heidelberg 2011, P. Jędrzejowicz et al. (Eds.), pp. 180–190.
12. Vis, I. F. A. Survey of Research in the Design and Control of Automated Guided Vehicle Systems, *European Journal of Operational Research*, 170, 2006, pp. 677–709.
13. Parulkar, G., Schmidt, D., Kraemer, E., Turner, J. and Kantawala, A., An architecture for monitoring, visualization and control of gigabit networks, *IEEE Network*, 11 (5) October 1997, pp. 34–43.
14. Walkowiak, T. and Mazurkiewicz, J. Algorithmic Approach to Vehicle Dispatching in Discrete Transportation Systems, In: *Technical approach to dependability* / eds Jarosław Sugier [at al]. Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej, 2010. pp. 173–188.
15. Walkowiak, T. and Mazurkiewicz, J. Availability of Discrete Transportation 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.
16. Walkowiak, T. and Mazurkiewicz, J. Functional Availability Analysis of Discrete Transport System Simulated by SSF Tool. In: *International Journal of Critical Computer-Based Systems*. 2010, Vol. 1, No 1–3, pp. 255–266.
17. Walkowiak, T. and Mazurkiewicz, J. Soft Computing Approach to Discrete Transport System Management, In: *Springer-Verlag, Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence*. 2010, vol. 6114, pp. 675–682.
18. Wooldrige, M., Agent-based software engineering, *Software Engineering*, IEEE Proceedings, Vol. 144 (1), February 1997, pp. 26–37.