Journal of Polish Safety and Reliability Association
Summer Safety and Reliability Seminars, Volume 3, Number 2, 2012

Mazurkiewicz Jacek

Nowak Katarzyna

Walkowiak Tomasz
Wrocław University of Technology, Poland

Fuzzy approach to complex system analysis

Keywords

dependability, modelling, complex systems, fuzzy logic

Abstract

The paper presents a new approach to reliability and functional analysis of sophisticated complex systems using fuzzy logic. We propose to use such methodology since dependability parameters of the system are mostly approximated by experts instead of classical sources of data. Presented approach show different idea – modeling system using the unified structure – in functional sense. We assess the reliability of the system by accumulated down time. Fuzzy logic based reliability analysis, as well as Computer Information System and Discrete Transport System modeling and simulation are presented. Moreover, results of numerical experiment performed on a test case scenario related to the economic and functional aspects using proposed methodology are given. Fuzzy approach allows reducing the problem of assumptions of reliability distributions and – this way – seems to be very interesting for real systems management and tuning.

1. Introduction

The global network of services and growing requirements of user satisfaction new infrastructures and new technologies become more and more popular, but also more complex. Complexity of this solutions force lower level of its description but in a same time high level perspective on what is going on in the system. Regardless of the level of abstraction many of parameters should be defined or measured to find the most accurate solution. Since some of them are uneasy to measure we propose to use know approach (fuzzy logic [11]) to model the system, since this kind of model can be very useful for further analysis of the system. In result we provide an approach or even an idea of the tool for network system administrator.

It is very important to note that complex system can be understood as several different service providers. Since service [9] is describe as a set of services components based on business logic, that can be loaded and repeatedly used for concrete business handling process (i.e. online service, transportation, etc.), therefore service can be seen as a set of service components and tasks, that are used to provide definite service in accordance with business logic for this process. On the high level importance of user request (task is defined by the user to the service located on the system) is the highest priority. For this reason we can assume and make a sweeping statement that user request is realized in the same matter in transportation network and in computer systems, but based on different resources (in first case - vehicles, in second - computers) [16].

In this paper we propose a combination of general view on a service network and it analysis using softcomputing methods. We call the approach as the functional-reliability models of network system exploitation. The computer systems analysis is the root for our elaboration but we believe it is useful for modelling of the wider spectrum of systems which realise tasks based on fully or partially available resources.

In this paper we shortly describe elements of transport system and computer system. Then some common elements are presented. Next sections provide the details of the fuzzy approach necessary to reliability and functional description and analysis of the computer information system as well as of the discrete transport system. The results show the essential practical data in the function of the reliability parameters of the system. We propose the parallel analysis of the computer system and the discrete transport system based on the unified fuzzy approach. Chapter ends with some general conclusion and remarks for the future works.

2. General complex system overview

2.1. Abstract approach to service network

The system is providing a service in a sense of user request accomplishment. As mentioned, in this paper, we speak about unified common approach, analyzing network services as a general network. Since the service is aim to realized user task, therefore the key point of the view is the *Task* (*T*) given to the systems, its specification and its time. Each task conditioned by the scenario – business logic, therefore choreography [6] within a service must be defined and known. To provide a task, we have to use system resources - *Technical Infrastructure* (*TI*) capable to realize specified choreography. Moreover, more than one choreography can be realized and more than one configuration of the service is possible. Since, choreography is based on predefined service components located in network nodes task should be seen as an input to the predefined *Business Service* (*BS*).
As mentioned, specifying the task and its parameters is a *user* role (*M*) and the time functions on each level of abstraction - C*hronicle of the System* (*K*). *Abstract Network Services* (*ANS*) can be represented as a 4-tuple [16]:

$$ANS =< T, BS, TI, M, K > \qquad (1)$$

This unified description can be realized in case of network system exploitation analysis. Based on (1) we will consider two networks with respect to low level view, that is: Computer Information System and Discrete Transport System.

2.2. Transportation systems

Discrete Transport System (*DTS*) and its tasks are seen differently than in a Computer System, but still there are based on the same schema. Task understood as a request to a system of transport resources (e.g. vehicles), transport infrastructure (e.g. roads) and a management system. For this reasons, we can speak about: *Task (T)* – as a several kinds of a commodity transported in the system, the commodities are addressed; *Business Service* (*BS*) – set of services based on business logic; each service component in *DTS* consists of a task of delivering a commodity from a source node to the destination one; *Technical Infrastructure* (*TI*) – infrastructure consisted of: *nodes* - to create the source, the destination and the trans-shipping points, *vehicles* - resources providing the service, described with some parameters, *roads* - paths of possible trip of vehicles, *Maintenance Crews; Client* (*CM*) - client allocated on one of nodes of the transport system and *Time-Table* (*TT*) – related to tasks and vehicles.
Then the model of *Discrete Transport System* can be described as follows [22]:

$$DTS = \langle TM, BS, TI, CM, TT \rangle \qquad (2)$$

2.3. Computer information system oriented to provide service

Task understood as a request to a service (i.e. online banking, ticket reservation, tax-payment) is specified by the user and executed in a *Computer Information System* defined as a farm of computers (i.e. servers) providing the service. With respect to its complexity such systems are called *Complex Information System* (*CIS*) – systems with extensive infrastructure aimed to satisfy user needs in case of a service based on computer network resources [18], [21]. Taking these aspects into consideration we focus on service and user requirements – functional and dependability. For this reasons, we can model *CIS* using the following elements: *Task* (*T*) – input data specified by the clients in case of business service usage; *Business service (BS)* – set of *service components* located on defined server that determined service possibilities, requirements and behaviour;
Technical infrastructure (HS) – network resources (devices and links) built to provide network service.
Chronicle of the system (*K)* – the time function on each level of abstraction; *Clients* (*M*) – set of users and its allocation, number of users of a given profile and their activities.
As mentioned since we propose to analyze *CIS* systems on a basis of their service, we can modify (1) and define *Complex Information System Oriented to Provide Service* as:

$$CIS =< Z, BS, HS, M, K > \qquad (3)$$

3. Discrete transport system (DTS)

3.1. System overview

Basic elements of system are as follow: store-houses of tradesperson, roads, vehicles, trans-shipping points, store-houses of addressee and transported media (commodities). The commodities are taken from store-houses of tradesperson and transported by vehicles to trans-shipping points. Other vehicles transport commodities from trans-shipping points to next trans-shipping points or to final store-houses of addressees. Moreover, in time of transportation vehicles dedicated to commodities could failed and then they are repaired (*Figure 1*) [22]. Different *commodities* are characterized by common attribute which can be used for their mutual comparison: capacity of commodities. The following assumptions related to the commodities are taken: it is possible to transport *n* different kinds of commodity in the system and each kind of commodity is measured by its capacity. *Road* is an ordered double of system elements. The first element must be a store-house of tradesperson or trans-shipping point, the second element must be a trans-shipping point or store-house of addressee. Moreover, the road is described by following parameters: length, number of maintain crews, number of vehicles moving on the road. The road is assumed to have no damages. A single *vehicle* transports commodities from start to end point of a single road, return journey realizes in an empty status and the whole cycle is repeated. The assumptions are as follow: a single kind of commodity is transported at the moment, vehicles are universal. The numerous vehicle parameters can be described as random variables using various distributions. The *store-house of tradesperson* is an infinity source of single kind of commodity. *Trans-shipping points* are a transition part of the system which is able to store the commodities. The trans-shipping point is described by following parameters: global capacity, initial state described by capacity vector of commodities stored when the system observation begins, delivery matrix. This matrix defines which road is chosen when each kind of commodity leaves the shipping point. The commodity could be routed to more than one direction. Only one vehicle can be unloaded at the moment. If the vehicle can be unloaded, the commodity is stored in the trans-shipping point. If not, the vehicle is waiting in the only one input queue serviced by FIFO algorithm. Only one vehicle can be loaded at the moment. If the vehicle can be loaded (i.e. the proper commodity is presented and it could be routed a given road) the state of trans-shipping is reduced. If not, the vehicle is waiting in the each output road FIFO queue [19]. The main task of the *store-houses of addressee* is to store the commodity as long as the medium is spent by the recipient. The store-house of addressee is described by following parameters: global capacity, initial state described as for the trans-shipping point, function or rule which describes how each kind of commodity is spent by recipients. Input algorithm is exactly the same as for trans-shipping point. Output algorithm can be described as: stochastic process, continuous deterministic or discrete deterministic one. Moreover, the following assumptions are taken: the capacity of the commodity can't be less than zero, "no commodity state” - is generated when there is a lack of required kind of commodity.

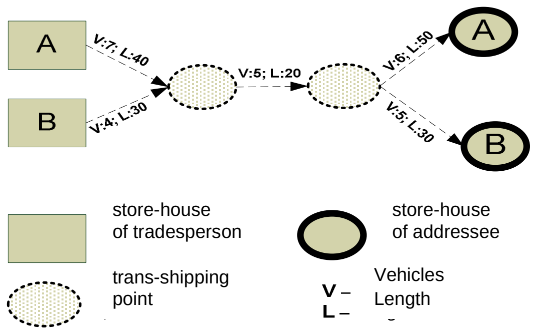

Figure 1. Discrete Transport System – case study

3.2. Economic analysis

The economic analysis is realized from vehicle owner's view-point. The revenue is proportional to number of store-houses of addressee, number of deliveries realized to single store-house of addressee and gain for single delivery to single store-house of addressee. Following costs are taken into account: penalty costs - paid by a transportation firm when there is a lack of commodity in the store-house of addressee, repair costs - proportional to a unit of repair time, vehicle usage costs - in a function of time (salary of drivers) and in a function of distance (i.e. costs of petrol). The economic quality of discrete transport system is described by overall gain function $G(T)$ estimated in given time-period T as difference between the revenue and costs. We have to remember that the overall gain $G(T)$ is a random variable [12].

4. Fuzzy approach to DTS

4.1. Case study

To show the possibilities of the proposed model we have analysed an exemplar transport network presented in *Figure 1*. It consists of two different commodities transported over network (marked as *A* and *B*) from two producers through two trans-shipping points to two consumers. Each commodity is spent by a given recipient. Roads lengths and the number of available vehicles are presented in *Figure 1*. All vehicles have the same parameters (see section 3.1.). To each road one maintains crew is assigned. Number of vehicles assigned to each road was calculated on a basis on required amount of commodities spent by each recipient taking into account some redundancy due to the fact of vehicle failures [11].

4.2. Fuzzy reliability parameters

We want to analyze the overall system gain *G(T)* in a function of fuzzy representation of truck reliability parameter: mean time of failures. We are not analyzing the classical reliability values: intensities of failures by its inverse of intensities since we think that it is much easier for expert to express the failure parameter in time units. For case study exemplar we assumed trapezoidal membership function for fuzzy representation of mean time of failure (let denote is as: $\mu_{M\mu}(m)$). The four trapezoidal parameters has been set to (500, 1000, 2000, 3000) hours. Assumption of the fuzzy membership function shape does not bound following analysis. One could use any other membership function and apply presented here methodology. Such system could be understood as a simple single input and single output fuzzy system. Applying fuzzy operator like max one could have achieved results. However, as it is stated in section 3.2., the overall gain is a random value. Therefore for a given system parameters (i.e. a given mean time repair time) we got a set of overall gain values [20].

4.3. Fuzzy gain representation

A classical way of presentation of random values is a mean value. In our case mean value of the overall gain. However, such method looses very important information from simulation results. Therefore, we propose to represent the gain *G(T)* as a fuzzy number. It could be done using for example the trapezoidal membership function. Its four parameters has been set based on mean value *m* and standard deviation *std* of achieved gain as (*m*-3*std*, *m*-*std*, *m*+*std*, *m*+3*std*). The results of fuzzy gain $\mu_G(g, m)$ (*g* spans possible values of gain, *m* spans possible values of mean time to failure) for case study system is presented in *Figure 2.* The results not satisfied the assumptions of fuzzy representation of mean time to failures (section 4.2.)

4.4. Fuzzy gain in function of fuzzy reliability

Having the fuzzy gain representation $\mu_G(g, m)$ for each crisp value of the fuzzy reliability parameter $\mu_{M\mu}(m)$, presented in *Figure 2*, we need to combine these two measures to achieve an overall fuzzy gain. We propose to apply max and multiply operator to solve this problem (results are presented in *Figure 3a*). It gives the following equation for the fuzzy gain membership function:

$$\mu_G(g) = \underset{m}{MAX}\{\mu_G(g,m)\cdot\mu_{M\mu}(m)\} \qquad (4)$$

where: *m* – mean time to failure, *g* – gain value.

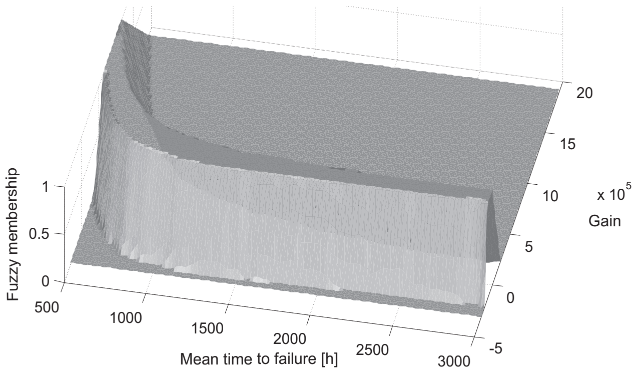

Figure 2. Fuzzy representation of overall gain in function of mean time to failure

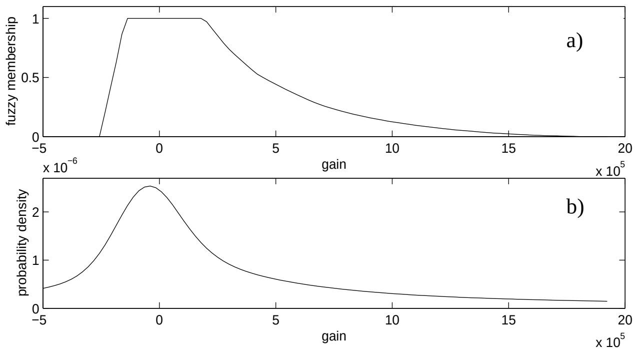

Figure 3. Results: the overall gain presentation by a) fuzzy method and b) probability density method

4.5. Probability density method

We propose also other way of final results presentation, based on probability density function estimation. Assuming that the fuzzy representation of mean time to failure (section 4.2.) is a way of stating the probability of vehicle mean time to failure, we could calculate the overall gain probability density function using slightly modified kernel method (with Gaussian kernels). The modification is done by multiplication each kernel by the weighted fuzzy trapezoidal function. Based on *N* results of overall gain g_i from computer simulation, calculated for given values of meant time to failure m_i, the density function *f(g)* is given by:

$$f(g) = \frac{1}{h\sum_{j=1}^{N}\mu_{M\mu}(m_j)} \cdot \sum_{i=1}^{N}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{g_i - g}{h}\right)^2\right)\cdot \mu_{M\mu}(m_i) \quad (5)$$

where h is a bandwidth parameter. It is set to optimal value based on maximal smoothing principle: *AMISE* - the asymptotic mean square error [17]. Results for the case study DTS are presented in *Figure 3b*.

5. Information systems

5.1. System description

The analyzed class of computer information systems is described on three levels. On the top level, it is represented by interacting service components.

At the bottom, technical level it is described by hosts, on which the services are located. The intermediate level describes the mapping between the other two. Service components (interacting applications) are responsible for providing responses to queries originating either from the clients or from other service components. While computing the responses, service components acquire data from other components by sending queries to them. The system comprises of a number of such components. The set of all services comprises a system. Communication between services works on top of Internet messaging protocols. The communication encompasses data exchange using the client-server paradigm. The over-all description of the interaction between the service components is determined by its choreography, i.e. scenarios of interactions that produce all the possible usages of the system. The service components interact with each other in accordance to the choreography. As the result, there are logical connections between service components. The service component is realized by some technical service, placed on some hosts. The assignment of each service components to technical components gives the system configuration.

5.2. Functional aspects

The performance of any information system has a big influence on the business service quality. It has been shown [9] that if user will not receive answer from the system in less than few seconds the user will resign from active interaction. Therefore, the most important part of the system model is an algorithm that allows calculating how long a user request will be processed be a system. Since, the processing of a user request is done by service components according to a given choreography, the overall processing time could be calculated as equal to time needed for communication between hosts used by each service component and the time of processing tasks required by each of service components. It is due to a fact that the processing time depends on the type of a task (its computational complexity), type of a host (its computational performance) on which a task is executed and a number of other tasks being executed in parallel. And this number is changing in a time during the system lifetime. Therefore, it is hard to use any of analytic methods to calculate the processing time. That is way we used the simulation approach that allows to monitor the number of executed tasks on each host during the simulation process. Having, calculated the time of processing a user request one could use it for assigning a request to be not correctly handled if it exceeds a time limit (10 seconds was used in presented experiments). There, could be also other sources of not correctly requests. The communication protocols (like HTTP) as well as Information services (for example JSP) have built-in timeouts. If a request is not finished within a given time limit (in most cases it could be set by one of configuration parameters) it is assumed to be failed. The other reason of not correctly handled requests in Information systems is a limit to a number of tasks handled by a technical service (i.e. Tomcat) at the same time. Since most of the user tasks consist of a sequence of requests, if one from the sequence fails the whole user request is assumed to be not correctly handled. The other sources of not correctly answered requests are hardware and software failures [2].

5.3. Fault discussion

The previous section introduced failures as a result of system functionality, i.e. a result of time-outs and maximum number of requests. We propose to extend failures to represents Information system faults which occur in a random way. Of course, there are numerous sources of faults in complex information systems. These encompass hardware malfunctions (transient and persistent), software bugs, human mistakes, viruses, exploitation of software vulnerabilities, malware proliferation, drainage type attacks on system and its infrastructure. We propose to model all of them from the point of view of resulting failure. We assume that system failures could be modelled a set of failures. Each failure is assigned to one of hosts and represents a separate working-failure stochastic process. The proposed fault model takes into account different types of faults, like: viruses, host and operating system failures. The occurrence of failure is described by a random process. The time to failure is modelled by the exponential distribution. Whereas the repair time by truncated normal distribution. We consider two types of failures for each host: with full dysfunction of host and 98% downgrade of host performance. The first one, represents the results of a host or operation system failure. The second type of faults (with 0.95 downgrade parameter) models a virus or malware occurrence [8].

6. Fuzzy approach to information system

6.1. Case study

To show the possibilities of the proposed model we have analyzed a system which consists of three networks: one is a client network, other service provider networks (secured by a Firewall). System is realizing simplified hotel booking system that allows booking an available apartment in hotel. Few servers are used for a proper service realization: *WebServer*, *HotelDatabase*, *ReservationServer*, *PaymentServerController*, *BackupWebServer* (*Figure 4*). The service and its choreography is described in Fig. 5. First of all, place of the hotel is being searched, than reservation is being made. At the end of this scenario payment is done with an interaction with given payment system [16].

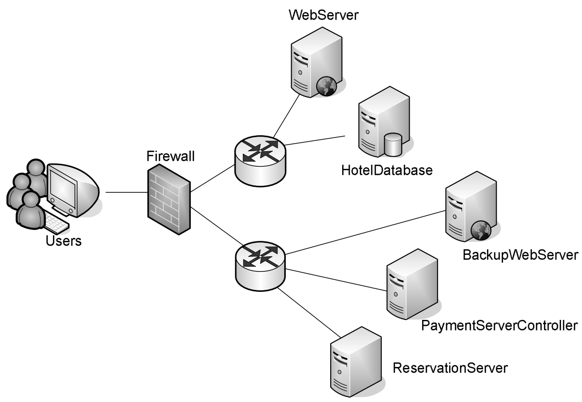

Figure 4. Information system – case study

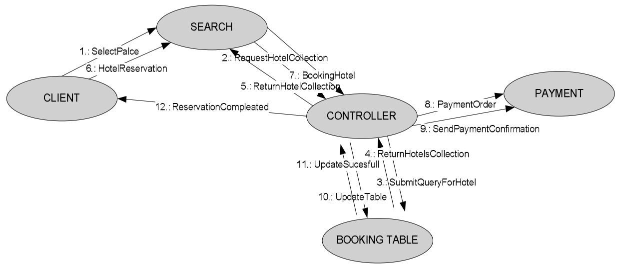

Figure 5. Information system choreography – case study

6.2. Fuzzy reliability parameters

We want to analyze the accumulated down time in a function of fuzzy representation of host reliability parameter: mean time of failures (*fh*) and mean repair time (*rh*). We are not analyzing the classical reliability values: intensities of failures but its inverse since we think that it is much easier for expert to express the failure parameter in time units. Moreover, we analyze the occurrence of virus and malware intrusions. It is described by mean time of virus occurrence (*fv*) and mean repair time (*rv*). We propose to use a trapezoidal membership function for fuzzy representation of mean time of failures and repair time for host and virus failures. Let note it is as: $\mu_{type}()$, where *type* is equal to *f_h*, *r_h*, *f_v*, *r_v* for mean time to host failure, host repair time, mean time to virus and virus repair time respectively. Assumption of the fuzzy membership function shape does not bound the analysis. One could use any other membership function and apply presented here methodology. For mean time of host failures, the four trapezoidal parameters of fuzzy membership function was set to (290, 330, 670, 710) days. Today's computer devices to not fail very often, that is way we consider a host failures mean time between one to two years. Faults that are related to viruses are more probable than a host failure, especially for systems that are exposed to attacks. Therefore, in our study mean time to virus occurrence fuzzy trapezoidal parameters were set to (100, 140, 340, 360). In case of repair time we use (4, 8, 32, 48) hours for host repair time and (2, 4, 16, 24) for virus repair time [12].

6.3. Fuzzy mean accumulated down time

In general we propose the analysis of mean accumulate down time in a function of reliability parameter:

$$MADT = \left(fh, rh, fv, rv\right) \quad (6)$$

MADT shows the number of hours a system is unavailable during a year. To define it, we need to state what does it mean that a information system is unavailable. We propose to use availability metric estimated as the ratio of number of requests correctly handled *($N_{OK}(t$ - Δ, t))* by the system over a number of all requests *($N(t$ - Δ, t))*, assuming a uniform rate of requests in time horizon (from *t* - *Δ* to *t*):

$$A(t) = \frac{N_{OK}(t-\Delta, t)}{N(t-\Delta, t)} \qquad (7)$$

The availability value is a random one, since it depends on number and types of failures that occur during analyzed year, therefore we define the expected value of a mean accumulated down time as follow:

$$MADT = E\left[\int_{t=0}^{1year} 1(A(t) < \theta)dt\right] \qquad (8)$$

The fuzzy approach to *MADT* creates a multiple input single output (*MISO*) system with four inputs. Applying fuzzy operator like max and multiply one could calculate the output membership function [11]:

$$\mu_{MADT}(t) = \underset{t=MADT(mh,rh,mv,rv)}{MAX}\left\{\mu_{f\_h}(fh)\cdot\mu_{r\_h}(rh)\cdot\mu_{f\_v}(fv)\cdot\mu_{r\_v}(rv)\right\} \qquad (9)$$

Results for the case study presented in *Figure 6*.

6.4. Fuzzy simplified approach

The calculation of formula (9) is complicated and time consuming. Therefore, we propose to calculate the output membership function based on L-R representation of fuzzy variables and approximation of resulting system fuzzy membership.

We calculate the resulting output function value (6) only for characteristic points of trapezoidal fuzzy membership of input values. For our four dimensional space, it gives $2^4 = 16$ points for fuzzy membership function equal to 1 and similarly for membership function equal to 0. Among each of these two groups, a minimum and maximum value of the output function values (6) is selected giving the resulting trapezoidal membership output function. Such representation guarantees a really simple and fast calculation of fuzzy output values. But this is only the rough approximation of fuzzy function of outputs values. The results for case testbed are shown in *Figure 6 b*.

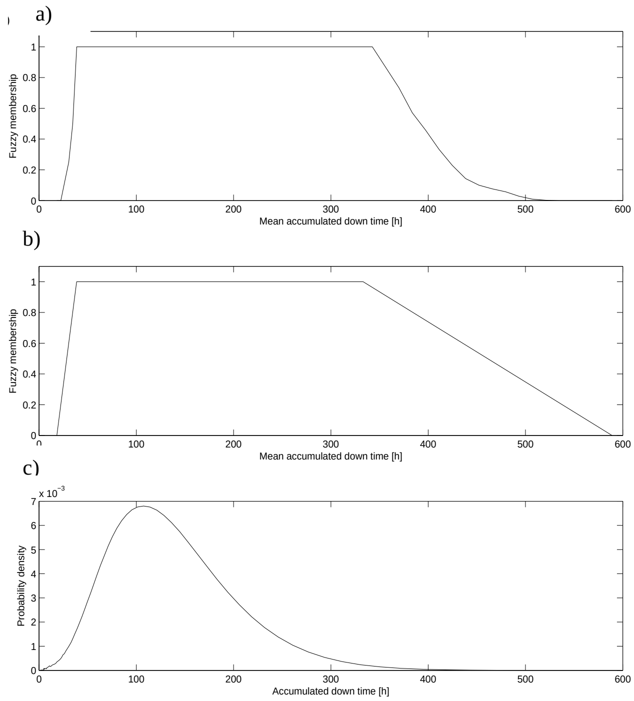

Figure 6. Results accumulated down time presentation: a) fuzzy method, b) fuzzy simplified method, c) probability density method

6.5. Probability density method

We propose also the other way of final results presentation, based on probability density function estimation. Assuming that the fuzzy representation of mean time to failure and repair time (section 6.2.) is a way of stating the probability of time to failure, we could calculate the overall gain probability density function using slightly modified kernel method (with Gaussian kernels). The modification is done by multiplication each kernel by the weighted fuzzy trapezoidal function. Based on *I* results of accumulated down time *ADT_i(fh, rh, fv, rv)* achieved for different reliability parameter values *(fh, rh, fv, rv)* by computer simulation, the density function *f(t)* could be approximated by [12]:

$$f(t) = \frac{1}{h\sqrt{2\pi}\sum_{j,k,l,m=1}^{J,K,L,M}\mu_{f\_h}(fh_j)\cdot\mu_{r\_h}(rh_k)\cdot\mu_{f\_v}(fv_l)\cdot\mu_{r_v}(rh_m)} \cdot \sum_{i,j,k,l,m=1}^{I,J,K,L,M}\exp\left(-\frac{1}{2}\left(\frac{ADT_i(fh_j,rh_k,fv_l,rv_m)-t}{h}\right)^2\right) \cdot \mu_{f\_h}(fh_j)\cdot\mu_{r\_h}(rh_k)\cdot\mu_{f\_v}(fv_l)\cdot\mu_{r_v}(rh_m) \qquad (10)$$

where *h* is a bandwidth parameter. It is set to optimal value based on maximal smoothing principle: *AMISE* - the asymptotic mean square error [17]. Results for the case study information systems are presented in *Figure 6 c*.

7. Conclusion

Summarizing, we proposed method of fuzzy analysis of complex systems. It is based on fuzzy reliability representation and computer simulation which allow to soften the typical assumptions related to the system structure and to reliability parameters of information system elements. Using proposed solution sophisticated systems can be verified against quality requirements, what makes this approach a powerful tool for increasing system dependability and by that increasing satisfaction of the service user. Considering complexity of the analyzed systems, we keep in mind that more and more parameters should be specified in a similarly manner. Researches in this area are still in progress, with respect to more complicated testbeds and larger data set. Moreover, it would be interested to model the changing number of the system users by fuzzy values.

The paper presents the abstract level of unified approach to complex system analysis based on fuzzy logic methodology. We show the promising results of applying the approach for the discrete transport system as well as for the computer information system. We think that presented approach could be a foundation for a new methodology of the reliability and quality analysis of various complex systems, which is much closer to the practice experience especially if the number of data to represent the reliability and / or functional parameters is insufficient or the values are not precise enough.

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] Barlow, R. & Proschan, F. (1996). *Mathematical Theory of Reliability*. Philadelphia: Society for Industrial and Applied Mathematics.

[2] Chan, K.S., Bishop, J., Steyn, J., Baresi, L. & Guinea, S. (2009). *A Fault Taxonomy for Web Service Composition*. In: Di Nitto, E., Ripeanu, M. (eds.) ICSOC 2007. LNCS Springer, Heidelberg, vol. 4907, 363–375.

[3] Conallen, J. (2000). *Building Web Applications with UML*. Addison Wesley Longman Publishing Co., Amsterdam.

[4] Davila-Nicanor, L. & Mejia-Alvarez, P. (2004). Reliability Improvement of Web-Based Software Applications. *Proceedings of Quality Software*, QSIC 2004, September 8-9, 180–188.

[5] Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag

[6] Hongli Y., Xiangpeng Z., Zongyan Q., Geguang P., & Shuling W. (2006). *A Formal Model for Web Service Choreography Description Language (WS-CDL)*, Proc. of ICWS 2006, IEEE Computer Society.

[7] Jennings, N. R. (2000). *On Agent-Based Software Engineering*. Artificial Intelligence 117, Elsevier Press, 277–296.

[8] Ma, L. & Tian, J. (2007). Web Error Classification and Analysis for Reliability Improvement. *Journal of Systems and Software* 80 (6), 795–804.

[9] Martinello, M., Kaaniche, M. & Kanoun, K. (2005). Web Service Availability–Impact of Error Recovery and Traffic Model. *Safety, Reliability Engineering & System Safety* 89 (1), 6–16.

[10] Mascal, C. M. & North, M.J. (2005). Tutorial on Agent-Based Modelling and Simulation. *Winter Simulation Conference.*

[11] Mazurkiewicz, J. & Walkowiak, T. (2003). Fuzzy Reliability Analysis. *ICNNSC 2002 – 6th International Conference Neural Networks and Soft Computing. Advances in Soft Computing*, 298–303. Physica-Verlag, Heidelberg.

[12] Mazurkiewicz, J., & Walkowiak, T. (2004). *Fuzzy Economic Analysis of Simulated Discrete Transport System.* Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI) Springer, Heidelberg, vol. 3070, 1161–1167.

[13] Mellouli, S., Moulin B., & Mineau, G. W. (2003). *Laying Down the Foundations of an Agent Modelling Methodology for Fault-Tolerant Multi-agent Systems*, ESAW, 275–293.

[14] Michalska, K., Mazurkiewicz, J. (2011). *Functional and Dependability Approach to Transport Services Using Modelling Language*. Springer LNCS/LNAI, LNAI 6923, Springer-Verlag Berlin Heidelberg, P. Jędrzejowicz et al. (Eds.), 180–190

[15] Nicol, D., Liu J., Liljenstam, M. & Guanhua, Y. (2003). Simulation of Large Scale Networks Using SSF. *Proceedings of the 2003 Winter Simulation Conference*, Vol. 1. 650–657.

[16] Nowak, K. & Mazurkiewicz, J. (2011). Multiagent Modeling and XML-Like Description of Discrete Transport System. *Transport and Telecommunication,* Editor: I. Kabashkin, Transport and Telecommunication Institute, Riga, Latvia, Vol. 12, No. 4, 14–26.

[17] Silverman, B.W. (1986). *Density Estimation*. Chapman and Hall, London

[18] Walkowiak, T. & Mazurkiewicz, J. (2009). Analysis of Critical Situations in Discrete Transport Systems. *Proceedings of International Conference on Dependability of Computer Systems,* Brunow, Poland, June 30–July 2, 2009. Los Alamitos: *IEEE Computer Society Press*. 364–371.

[19] Walkowiak, T., & Mazurkiewicz, J. (2008). Availability of Discrete Transport System Simulated by SSF Tool. *Proceedings of International Conference on Dependability of Computer Systems*, Szklarska Poreba, Poland, June, 2008. Los Alamitos: IEEE Computer Society Press, 430–437.

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

[21] Walkowiak, T., & Mazurkiewicz, J. (2010). *Algorithmic Approach to Vehicle Dispatching in Discrete Transport Systems*. Technical approach to dependability / Ed. by Jarosław Sugier, et al. Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej. 173–188.

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