Proceedings of the 11th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'11), 19–22 October 2011, Riga, Latvia, p. 390-397. ISBN 978-9984-818-46-7
Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia

WORKLOAD MODELLING FOR COMPLEX INFORMATION SYSTEMS

Katarzyna Michalska

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

The paper deals with a novel approach to modelling of a workload for Complex Information Systems. Proposition of a formal model of Complex Information Systems is formulated. Next workload model that based on user behaviour is presented. Using XML format User Description Language was formulated. Proposed language is described in details with a shortly described exemplary file. This paper focuses on a modelling of the user, but some references for further usage of this approach is introduced. Proposed approach can be used as a tool helpful to administrate the system.

Keywords: workload, complex system, modelling, user model

1. Introduction

The Internet environment has changed significantly in the last decades. With a growing population and usage of Internet, evaluation of information systems becomes more and more important. Studies included many types of computer analysis and emulation. It entails an increase of modelling concept and new techniques of system description. Large field of the computer science studies is considering problems that these systems faced. Many techniques and metrics [3,21,28] are proposed to increase administrator or designer knowledge about the system and its behaviour, since its complexity is still a crucial point of its management. Every of those mentioned aspect is done in one more purpose – user satisfaction. Since management of the system is done not only to provide a service in a system, but mainly to satisfy user requirements for the service that is realized in the system infrastructure. For this reasons, not only system infrastructure (software and hardware resources) should be examined, but also a workload that the system receive as an input – user actions. Within last few years, big and small companies that provide this kind of services have a difficult task to do – fulfil still growing/increasing user requirements/demands. To do that, many researchers have been done not only to name the issues that are required but also to measure them in the sense of their usability and functionality. For example it has been proven [18] that if user will not receive answer for the service in less than 10 seconds he/she will probably resign from active interaction with the service and will be distracted by other ones. That is why it is so important to provide maximum service response time between 0,1-8 seconds.

In this paper, the system workload will be understood as a load generated by all users of the Complex Information System (defined in section 2). Next, user modelling possibilities will be examined and an authors approach will be presented (section 3). Possibilities of language usage are presented in section 4, that conclude works related to the topic of the paper.

2. Complex Information System

Modern information systems can be seen as a complex ones, since Complex Information Systems (CIS) are understood as dynamic systems with numerous components and interconnections, interactions or interdependencies which are difficult to describe, understand, predict, manage, design, and/or change [2]. In connection with modelling and control complexity, complex systems have specific characteristics, among which are the following:

- uniqueness;
- weak structure of knowledge about the system;
- the composite nature of system;
- heterogeneity of elements composing the system;
- the ambiguity of factors affecting the system;
- multivariation of system behaviour;
- multicriteria nature of estimations of system's properties; and, as a rule,
- high dimensionality of the system.

Complex Information System can be as well large-scale or a small scale depending of the system proposes. In this paper, we will examine user actions in the system, therefore, we propose to analyse the CIS from the business service point of view.
Generally speaking users of information system are generating tasks which are being realized by the IS. 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. This system can be model as a 4-tuple [22]:

$$CIS = \langle Client, BS, TI, Conf \rangle \tag{1}$$

$Client$ – finite set of clients,
BS – business service, a finite set of service components,
TI – technical infrastructure,
$Conf$ – information system configuration.

Business service (BS) is a set of services based on business logic, that can be loaded and repeatedly used for concrete business handling process (i.e. ticketing service, banking, VoIP, etc). 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 which consists of a set of tasks. Tasks are the lowest level observable entities in the modelled system. It can be seen as a request and response form one service component to another [33].

Technical infrastructure (TI) is considered as a set of hosts and computer network and is assumed to have the aspects of TCP/IP traffic are negligible. Each host is described by server name (unique ID), host performance parameter and a set of technical services (i.e. apache web server, MySQL database).

Clients – consists of a set of users where each user is defined by its allocation (host), number of concurrently ruing users of given type, set of activities (a sequence of task calls – name of task and a name of service component) and inter-activity delay time.

System configuration (*Conf*) is a function that gives the assignments of each service components to a technical service and therefore to hosts since a technical set is placed on a given host.

3. User Behaviour Model

As mentioned before in every computer information system, one of the most important concern is not only the infrastructure of the system (software and hardware that realized tasks in the system), but also users, that are determining its behaviour. When we look at the system from a higher perspective, it is a user that in every system is at the top of the hierarchy since it provides network load sequence. The term *user modelling* or *user behaviour model* has been applied to this process of gathering information about the users of a computer system and of using the information to provide services or information adapted to the specific requirements of individual users (or groups of users). In some articles [7] this information's are called *user profile*. These models will catch the sequence of user interactions at a higher level. Such models will in general be hierarchical, as the workload generated at a higher level will result in a stream of workload requests at a lower level. Examples for hierarchical user models can be found in [12] and [13]. In [15] another popular method for catching inter-command dependencies is to apply Markov

Chains for user behaviour modelling can be found. More approaches that are static can be found in [30] and [4], where users are described by *n*-dimensional vectors.

Non-formalizable problems and impossibility to strict mathematical formulation of those problems led to creation a user's behaviour model in more abstract way. Based on the psychological [18] results this model is accompanied by expert knowledge and information stored in the system log files. Still this knowledge is not standardized and strongly depended on an expert intuition. Therefore, we propose to look or create standards description of the user, that can be further used for analysis.
Unfortunately, in literature the exit not known user description language, since analysis based on a user model is mainly based on an input model of the tool (i.e. simulation input file). There exists many organizations involved in creating and standardize languages describing various aspect of computer systems. These are as follows:

- OASIS (Organization for the Advancement of Structured Information Standards) [26],
- BPMI (Business Process Management Initiative) [5],
- OMG (Object Management Group) [23],
- W3C (World Wide Web Consortium) [34],
- WfMC (Workflow Management Coalition) [35],
- OAGi (Open Applications Group) [25].

Within work that is done by those groups we can find some models that in some sense contains the information about user (System Description Language [1, 27], Web Service Choreography Description Language [14], Business Process Modelling Notation [5], etc. [34]), but they are mainly focus on the system or service behaviour after user request. Some papers propose to use UML diagrams, as representation of as this model, but it cannot be used directly as an input model for analysis. Additionally description language for this system should be as close as real, not only to a mathematical description of the system, but to real system behaviour and its parameters. For this reasons, we propose an authors solution of a description language for a proposed model, called UDL (User Description Language). As a format of the proposed language XML (Extensible Mark-up Language) was chosen. Main reason is a simple (easy to learn) and readable structure, that can be easily converted to text or other format. Moreover, XML is supported not only with various tools (providing validation possibilities) but is also supported by many programming languages and framework in case of quicker and more efficient implementation. Figures 1 and 2 show parts of the XML Schema for User Description Language. As can be seen, each element of the system described in section is modelled as a complex element with appropriate sub-elements and attributes. Expect easiness of potential soft computing analysis, promising scalability and portability (between analysis tools) can be named as the main advantage of the language usage. Moreover, the language is easy to read and to process using popular and open-source XML tools. The only disadvantage of the format is a large size of the file because of the metadata usage.

3.1. User Description Language – schema

The model can be divided into two parts:

- client general description
- action description.

First one (Fig.1) defines *ClientID*, and parameters connected with time i.e. *ClientStart*, which define point in time, that the user will start its interaction with a system. Please note that, request to the system is not coming at the beginning of system, since user is defining its requests in different time slots, that can be modelled as a fixed values or as a random values using some distribution.

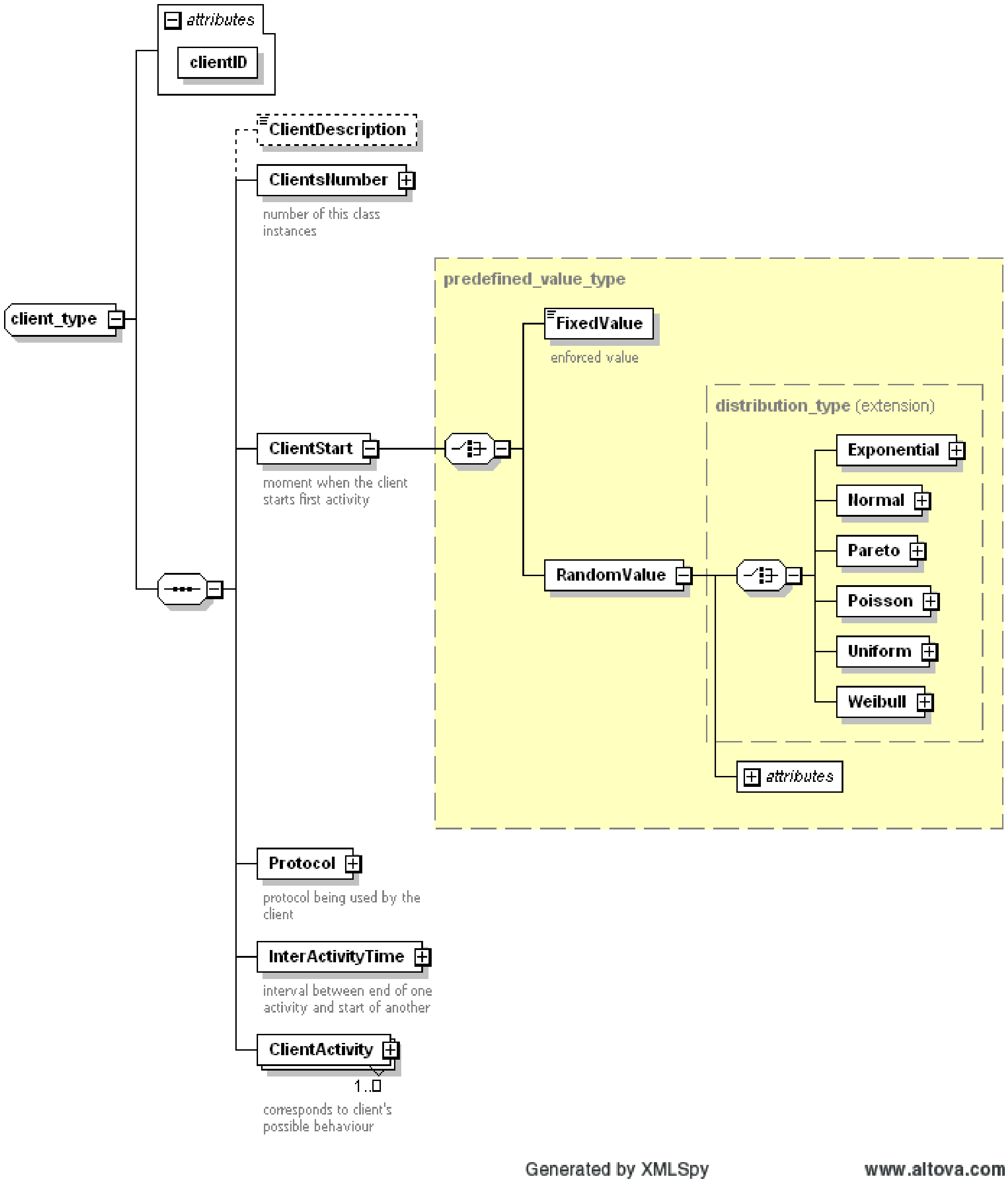

Figure 1. User Description Language – general user profile

As it is mentioned, users can have various profiles representing a group of task or common characteristics. For these reasons, *ClientsNumber* parameters can be defined as a number of users representing some profile. Furthermore, each profile can have common chain of activities (tasks for the system). Those activities will be realized in the system using technical infrastructure of the system. Each user need some time, to think about next step of its action, therefore *InterActionTime* parameter was proposed, and modelled as a distributed value. Let us now imagine that we have different client's sessions. One will be browsing simple Web Page, the other Web Page connected with database. Knowing that session needs to be authorized or waiting for Web Page response, user will have more patient, then or these reasons, *PatienceTime* was set as one of the model parameter. Additionally, we can add some conditions connected with a chain (*PreCondition*) of requests, but in many cases, the precondition is known from service choreography.

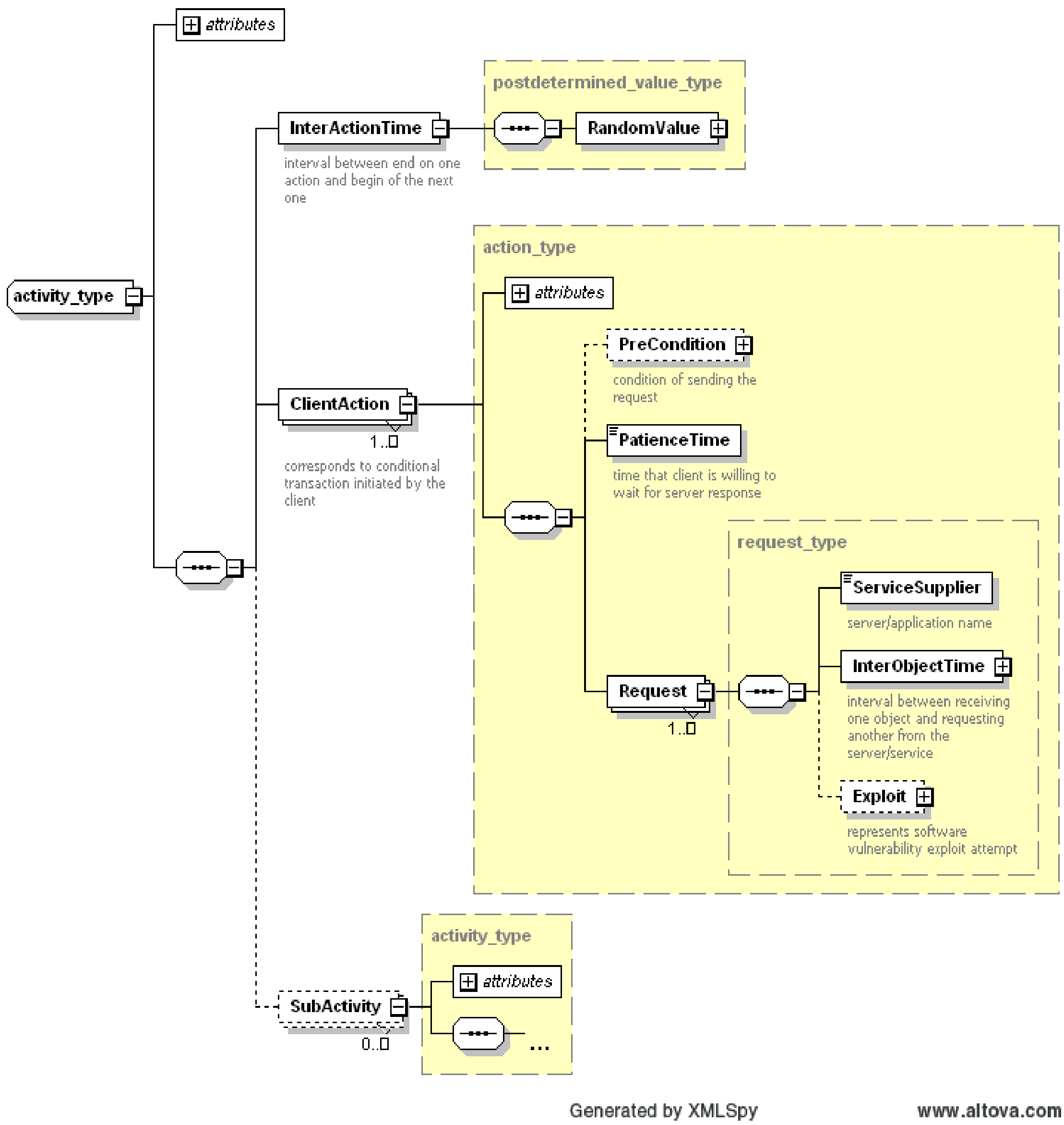

Figure 2. User Description Language – action description

3.2. User Description Language – sample file

Let's now look at a sample file describing user "client1". As a typical user (i.e. Alice) it creates profile for 50 clients. Staring time for the client will have random value described by exponential destitution (lambda equal to 0.1). Assuming that the service is Job Search Portal with a possibility to see some news about jobs, then client can have two activities: *JobSearch* (done in an 80% of the cases) and *NewsSearch* – made by 20% of the clients. Please note that in this example, user is aware that he/she is filtering some job profiles database, so its *PatienceTime* is longer than in case of browsing simple Web Page. This assumption can be changed easily, by making simple change in a UDL file (see Fig.3).

```
<Client clientID="client1">
                <ClientDescription>typical client (Alice)
        </ClientDescription>
                <ClientsNumber>
                        <FixedValue>50</FixedValue>
                </ClientsNumber>
                <ClientStart>
                        <RandomValue uncertain="false">
                                <Exponential lambda="0.1"/>
                        </RandomValue>
                </ClientStart>
                <Protocol>
                        <HTTP headerSize="1000" persistent="false"/>
                </Protocol>
                <InterActivityTime>
                        <RandomValue uncertain="true">
                                <Exponential lambda="0.3"/>
                        </RandomValue>
                </InterActivityTime>
                <ClientActivity activityName="JobSearch"
                        occurenceFrequency="70">
                        <InterActionTime>
                                <RandomValue uncertain="true">
                                        <Exponential lambda="0.1"/>
                                </RandomValue>
                        </InterActionTime>
                        <ClientAction actionType="standard request"
                        actionName="lookJobs">
                                <PreCondition/>
                                <PatienceTime>25</PatienceTime>
                                <Request>
                                        <ServiceSupplier>web_server
                                        </ServiceSupplier>
                                        <InterObjectTime>
                                                <RandomValue uncertain="true">
                                                        <Exponential lambda="0.1"/>
                                                </RandomValue>
                                        </InterObjectTime>
                                </Request>
                        </ClientAction>
                </ClientActivity>
                <ClientActivity activityName="NewsSearch"
                                occurenceFrequency="30">
                        <InterActionTime>
                                <RandomValue uncertain="true">
                                        <Exponential lambda="0.1"/>
                                </RandomValue>
                        </InterActionTime>
                        <ClientAction actionType="standard request"
                                actionName="lookNews">
                                <PreCondition/>
                                <PatienceTime>15</PatienceTime>
                                <Request>
                                        <ServiceSupplier>web_server
                                        </ServiceSupplier>
                                        <InterObjectTime>
                                                <RandomValue uncertain="true">
                                                        <Exponential lambda="0.1"/>
                                                </RandomValue>
                                        </InterObjectTime>
                                </Request>
                        </ClientAction>
                </ClientActivity>
        </Client>
```

Figure 3. User Description Language – sample

4. Conclusions

The purpose of the workload model we derived is to generate web traffic for analysis studies. Constructed general model and language can be now used as an input for such tool. Since, Complex Information System is mostly examined using network simulator environment. We studied several network simulation packages and libraries to determine how web traffic is generated in these packages. We considered only open source software because of the possibility of extending open source packages with our traffic model. We looked at the REAL, Omnet++, NS packages and SSFNet simulator [10,19,20,24,29,31]. For the purpose of already ongoing research proposed in [22,33] the SSFNet tool was chosen. The event-simulation used in this solutions is based on the Scalable Simulation Framework (SSF) [31], which is used for SSFNet [32] computer network simulator. For the purpose of simulating service-based systems we have used Parallel Real-time Immersive Modeling Environment (PRIME) [16, 17] implementation of SSF due to a much better documentation available for the original SSF. We have developed a generic class derived from SSF Entity, which is a base of classes modelling system objects. It is important to notice that the Monte-Carlo approach is used [11]. The original SSF was not designed for this purpose so some changes in SSF core were done to allow restarting the simulation from time zero several times within one run of simulation programme.

Presented in this paper model and language was widely used during DESEREC (Dependability and Security by Enhanced Reconfigurability) [9] project that author has been strongly involved. Within the project UDL language was used to model user behaviour. Expert teams involved in the project, as much as simulator development team, improved its format.

Acknowledgment

The presented work was funded by the Polish National Science Centre under contract no. 4759/B/TO2/2011/40.

References

1. Aime, M., Atzeni, A., Pomi, P. AMBRA – Automated Model-Based Risk Analysis, *The 3rd International Workshop on Quality of Protection (QoP 2007) (Affiliated with ACM CCS) Alexandria, VA,* USA, October 29, 2007.
2. Bagdasaryan, A. Discrete dynamic simulation models and techniques for complex control systems, *Simulation Modelling Practice and Theory,* Vol. 19, Issue 4, April 2011, pp. 1061-1087.
3. Bagrij, L., Michalska, K., Woda, M. Dependability Evaluation of Service-oriented Information System Using Computer Simulation and Post-processing Results. *Proceedings of the International Conference on Dependability of Computer Systems. DepCoS – RELCOMEX 2008.* IEEE Computer Society [Press], 2008. (Accepted for publication).
4. Bar-Yam, Y. *Dynamics of Complex Systems*, Reading, MA: Addison-Wesley, 1997.
5. BPMN (*Business Process Modelling Notation*); *http://www.bpmn.org/*
6. Business Process Management Initiative; *http://www.bpmi.org/*
7. Calzarossa, M., Haring, G., Serazzi, G. Workload modeling for computer networks. In: *Architektur und Betrieb von Rechensystemen / U. Kastens and F. Ramming, Eds.* Berlin: Springer-Verlag, 1988, pp. 324-339.
8. Calzarossa, M., Massari, L., Tessera, D. Workload Characterization Issues and Methodologies. In: *Performance Evaluation: Origins and Directions: Lect. Notes Comput. Sci.,* 2000, pp. 459-482.
9. DESEREC: DESEREC Project Home Page. http://www.deserec.eu/
10. Duflos, S., Diallo, A. A., Le Grand, G. An Overlay Simulator for Interdependent Critical Information Infrastructures. *Dependability of Computer Systems, DepCoS-RELCOMEX '07.* IEEE Press, 2007, pp. 27-34.
11. Fishman, G., *Monte Carlo: Concepts, Algorithms, and Applications*. New York: Springer-Verlag, 1996.
12. Haring, G., Fundamental principles of hierarchical workload description. In: *Workload Characterization of Computer System and Computer Networks.* North-Holland, Amsterdam, 1986, pp. 101-111
13. Hlavacs, H., Hotop, E., Kotsis, G. Workload Generation By Modeling User Behavior. *Proceedings of OPNETWORK 2000, August 28 – September 1, Washington, USA.*

14. Hongli,Y., Xiangpeng, Z., Zongyan, Q., Geguang, P., Shuling, W. A formal Model for Web Service Choreography Description Language (WS-CDL). *Proc. of ICWS 2006.* IEEE Computer Society, 2006.
15. McTear, M. F. User modelling for adaptive computer systems: a survey of recent developments, *Artificial Intelligence Review*, Vol. 7, No 3-4, 1986, pp. 157-184.
16. Nicol, D., Goldsby, M., Johnson, M. Fluid-based simulation of communication networks using SSF. *1999 SCS European Simulation Conference*. Germany: Erlangen, 1999.
17. Nicol, D., Liu, J., Liljenstam, M., Guanhua, Y. Simulation of large scale networks using SSF. *Proceedings of the Winter Simulation Conference (2003). Volume 1, 7-10 December 2003*, pp. 650-657.
18. Nielsen, J., *Usability Engineering*. San Francisco: Morgan Kaufmann, 1994.
19. Nikolaidou, M., Anagnostopoulos, D. A distribution system simulation modelling approach, *Simulation Modelling Practice and Theory*, Vol. 11, 2003, pp. 251-267.
20. NS-2 Simulator Home Page, *www.isi.edu.nsnam/nas/*
21. Michalska, K., Walkowiak, T. Modelling and simulation for dependability analysis of information systems. *Information systems architecture and technology. Model based decisions, Ofi-cyna Wydaw. PWroc.* Wroclaw, 2008, pp. 115-125.
22. Michalska, K. & Walkowiak, T. Simulation approach to performance analysis information systems with load balancer. *Information systems architecture and technology: advances in Web-Age Information Systems.* 2009, pp. 269-278.
23. Object Management Group; http://www.omg.org/
24. OMNeT++ Simulator Home Page, *www.omnetpp.org*
25. Open Applications Group; *http://www.openapplications.org/*
26. Organization for the Advancement of Structured Information Standards; *http://www.oasis-open.org/*
27. POSITIF: POSITIF Project Home Page. http://www.positif.org/
28. Puigjaner, R. Performance Modelling of Computer Networks. In: *Proceedings of the 2003 IFIP/ACM Latin America conference on Towards a Latin American agenda for network research (LANC '03). ACM, New York.* NY, USA, pp. 106-123.
29. Rahman, M., Pakštas, A., Wang, F. Network modelling and simulation tools, *Simulation Modelling Practice and Theory,* Vol. 17, Issue 6, July 2009, pp. 1011-1031.
30. Raghavan, S.V., Vasukiammaiyar, D., Haring, G. Hierarchical approach to building generative networkload models, *Computer Networks and ISDN Systems*, Vol. 27, No 1, 1995, pp. 1193-1206.
31. Riley, G. Large-scale network simulations with GTNetS. In: *Proceedings of the 2003 Winter Simulation Conference 2003, Volume 1, 7-10 December 2003,* pp. 676-684.
32. SSFNet Simulator Home Page, *www.ssfnet.org*
33. Walkowiak, T. *Information systems performance analysis using task-level simulator, DepCoS - RELCOMEX 2009, IEEE Computer Society Press,* 2009, pp. 218-225.
34. Walkowiak, T., Michalska, K. Simulation approach to performance analysis information systems with load balancer. In: *Information systems architecture and technology: advances in Web-Age Information Systems / Ed. by Leszek Borzemski [i in.].* Wrocław : Oficyna Wydawnicza Politechniki Wrocławskiej, 2009, pp. 269-278.
35. World Wide Web Consortium*; http://www.w3.org/*
36. Workflow Management Coalition; *http://www.wfmc.org/*