ONTOLOGICAL APPROACH TO A CONSTRUCTION OF THE SIMULATION SYSTEM FOR THE SPECIFIC DOMAIN

Elena Zamyatina, Alexander Mikov, Roman Mikheev

***Abstract**: This paper discusses the problem of simulation system constructing for the specific domain. Authors suggest using the ontological approach. The simulation system TriadNS is considered. This simulation system is dedicated for computer networks design and analysis. Authors represent base ontology and some other ontologies describing the concepts of a specific domain.*

***Keywords**: simulation, ontology, computer network, multi model approach*

***ACM Classification Keywords**: I.6 Simulation and Modeling, I.6.8 Types of Simulation - Distributed: I.2 Artificial Intelligence, I.2.5 Programming Languages and Software - Expert system tools and techniques*

Introduction

Simulation system is wide spread and well known method used to analyze the complicated dynamic systems (to design and to analyze the computer networks, or computing systems, or logistic systems, for example). Very often this method may be the only one for the investigations (if it does not have any analytical decisions). But the use of simulation may be more effective if the specialists in particular domains take part in these investigations. Indeed it is very rational to use graph theory to study computer networks (path finding). The traffic exploration may be carried out with the help of queueing theory; in order to explore the parallel and distributed systems it often requires to know the theory of Petri Nets. So it is expedient to make the usual (standard) terms and concepts available for the investigator in a specific domain. Ontology is a very convenient tool here because it consists of a set of concepts and a set of a links between these concepts.

The paper considers one more approach. It is based on the language workbench. Moreover, if it is necessary to investigate a problem, situation or other complicate systems more precisely one has to create different models and to transform these models [Sokolov & Yusupov, 2005]. So an investigator must work with the set of models. The set of models allows improving the quality of the investigations and makes simulation experiment more efficient (qualimetry of simulation model).

Ontologies in simulation and simulation systems

Ontology is the description of types of entities of a specific domain, their attributes and relations between the entities. One can describe every specific domain (as a part of real world) by ontologies. Ontologies are created and used in various specific domains, and it is well known how to use the ontologies in the simulation.

But the creation of ontologies for simulation is a rather complicated problem and it is not easy to solve it because the simulation may be applied to various specific domains (chemistry, physics, logistic, transport and etc.). Moreover simulation is based on the mathematics, statistics, and the theory of probability (and so on), so their ontologies have to be the basic ones.

Ontologies may be used on the various stages of simulation beginning from the stage of data collection onwards (data about a specific domain) and accomplishing with the stage of validation [Sargent, 2005]. One can consider some ontology driven program tools dedicated to simulation (High Level Architecture, for example) and the problem of the alliance of the federates (in HLA) [Rathnam & Paredis, 2004]. This approach uses ontologies to describe the demands to which the federator's interface must correspond; what is needed for their successful cooperation. Moreover the ontologies are used to construct these demands with due regard to particular domain.

The paper [Liang & Pardis, 2003] presents the ontology of ports. This ontology can be considered as a mean for automatic building of a simulation model containing several components. Ports describe the interfaces of the components of a model or the interfaces of the subsystems in the system configuration. Overall system may be considered as the several subsystems or components connected to each other via well-defined interfaces. Ontologies are used successfully in other cases and this fact is reflected in various papers [Benjamen et al, 2005; Benjamen et al, 2006].

Setting to the specific domain

So it is necessary to set a simulation system to the specific domain. In fact it is the description of a simulation model with the help of specific concepts and relations between the concepts. And if an investigator uses graphical editor it is possible to describe a simulation model with the help of one of visual languages.

It is possible to use special program tools of language workbenches (or DSM-platforms), dedicated to creation of DSL (Domain Specific Language), to the setting simulation system to the specific domain. The second approach – program tools, based on ontologies. Let us consider the first approach.

Language workbench and DSL creation

There are various language workbenches (MetaEdit+, DSL Tools and so on) allowing to create meta models (DSL). These meta models permit to create models of a specific domain. Meta-language [Сухов, 2013] is one of the language workbenches and it was designed and implemented in Perm State University. Designers of meta-language removed the shortcomings of similar program tools. The design of simulation system set to the specific domain consists of several steps:

- It is creation of a new model, every model has a name and sometimes it is necessary to describe the model's attributes (if required). Meta-language has a graphical editor for model building. It is necessary to define the basic construction of the language. The basic elements in Metalanguage are entities, relations, restrictions. So an investigator defines basic entities of a meta model (DSL), relations between these entities and imposes a restrictions of the meta models;
- Afterwards an investigator creates models containing particular entities of specific domain and relations between these entities with the help of DSL;
- It is necessary check the validity of a model, the investigator has to be sure that the model satisfies all restrictions. If some of these restrictions are violated the investigator may be informed about this fact.

Program tools mentioned above may be applied not only for setting a simulation model to a specific domain and be applied to particular simulation system. These program tools have a wider application; the main application is the business model creation [Zamyatina et al, 2013].

Setting to specific domain using ontological approach

The second approach of simulation model setting to a specific domain is model-oriented approach, based on the ontologies. This approach is used, for example, to build a variety of GUI software [Gribova & Cherkezshvili, 2010].

The setting of simulation system to a specific domain may be done with the help of the ontologies for the following reasons:

- TriadNS has basic ontology and ontologies which are needed for the setting to a specific domain. Moreover ontologies may be needed for the completeness of partly defined simulation model. The complete model may be built due to including the program fragments (routines) into simulation model. Special program component TriadBuilder searches these fragments with the help of ontology in data base of routines using semantic types;
- It is well known that computer networks are an example of a rapidly changing specific domain (new types of networks, new protocols, new algorithms of message exchange and new routing algorithms appear). It is therefore necessary to have special linguistic and software tools

allowing to describe these new devices and algorithms and so to perform an automatic or semiautomatic setting to a specific dynamically changing domain;

- Computer networks can include a large number of computing nodes, so the simulation requires a large amount of time. The way out is the parallelization of a simulation run, distribution of objects of the simulation model by the nodes in intensive computing environment. At that, it is necessary to keep the causality of events;
- It is important to optimize simulation system in the respect of time because computer networks may consist of tremendous number of nodes. So it is necessary to use the resources of several computing nodes of mainframe, or cluster (and etc.) in order to reduce the overall time of simulation experiment. Simulation model in this case must be presented as a set of logical processes in this case. These processes are distributed between computing nodes usually. Processes communicate one another during simulation run, classical synchronization algorithms (optimistic and conservative) govern the functioning of logical processes. Efforts of many authors focus on improving these algorithms. In TriadNS also developed improved synchronization algorithms (based on the optimistic one). Feature of these algorithms is that they use the knowledge of the model, and this knowledge is partially extracted from the ontology. Moreover, it is necessary to have equal workloads on the computing nodes during simulation run. Special component of simulator named Triadbalance provides load balancing during simulation experiment. It is designed as multi-agent systems. Agents act according to the rules extracted from ontologies both basic and associated with particular subject area [Mikov et al, 2013];
- Simulation model verification and validation may be carried out using ontologies too [Zamyatina et al, 2013]. Ontologies contain the rules and restrictions. Simulation model has to be built in accordance with these rules and restrictions;
- It is convenient to integrate components automatically into simulation model or to include the new components from the other simulation models and systems into a simulation model.

So we can conclude that ontologies may be used at every stage of simulation of computer network. So it is rational to use them for setting to a specific domain. Investigations show that ontologies allow making simulation system adaptable and providing it is openness [Zamyatina & Mikov, 2012].

Let us consider the principles of simulation model construction in TriadNS, linguistic and software tools for collecting data during simulation run and etc. before the precisely consideration of ontologies in TriadNS.

Simulation model representation in TradNS

Computer simulator was designed and implemented on the base of CAD TRIAD [Mikov, 1995].

Simulation model in TriadNS is represented by several objects functioning according to some scenario and interacting with one another by sending messages. So simulation model is μ={STR, ROUT, MES} and it consists of three layers, where STR is a layer of structures, ROUT – a layer of routines and MES – a layer of messages appropriately. The layer of structure is dedicated to describe objects and their interconnections, but the layer of routines presents their behavior. Each object can send a message to another object. So, each object has the input and output poles (P_{in} – input poles are used to send the messages, P_{out} – output poles serve to receive the messages). One level of the structure is presented by graph P = {U, V, W}. P-graph is named as graph with poles. A set of nodes V presents a set of programming objects, W – a set of connections between them, U – a set of external poles. The internal poles are used for information exchange within the same structure level; in contrast, the set of external poles serves to send messages to the objects situated on higher or underlying levels of description. Special statement <message> through <name of pole> is used to send the messages.

One can describe the structure of a system to be simulated using such a linguistic construction:

structure <name of structure> ***def*** (<a list of generic parameters>) (<a list of input and output parameters>) <a list of variables description> <statements>) ***endstr***

Special algorithm (named “routine”) defines the behavior of an object. It is associated with particular node of graph P = {U, V, W}. Each routine is specified by a set of events (E-set), the linearly ordered set of time moments (T-set), and a set of states {Q-set}. State is specified by the local variable values. Local variables are defined in routine. The state is changed if an event occurs only. One event schedules another event. Routine (as an object) has input and output poles (Pr_{in} and Pr_{out}). An input pole serves to receive messages, output – to send them. One can pick out input event e_{in}. All the input poles are processed by an input event, an output poles – by the other (usual) event. The special statement ***out*** (***out*** <сообщение> ***through*** <имя полюса>) serves for sending a message.

A set of routines defines a routine’s layer ROUT.

The layer of messages (*MES*) is necessary for a description of a message with complicate structure.

Simulation algorithm is a set of the objects which function in accordance to their scenarios, objects send messages to one another and they are governed with the help of special program – “simulator”. Each routine has local calendar of the scheduled events. Each scheduled event has a time stamp (a scheduled moment of time when event will occur). Simulator has to find the smaller time stamp (among all local calendars) and call the appropriate routine containing the event with the smaller time stamp.

But it is very important to assess the simulation model and it is behavior in defined conditions and in accordance to appropriate restrictions.

Data Collection and Processing of Data During simulation run

Special program component – the *condition of simulation* defines the scenario of simulation experiment, the criterions of simulation run termination, a list of simulation model elements (the variables, the events, the input and output messages) which are have to be examined and processed during simulation run with the help of the *information procedures* and the scenarios of the completing processing of the results of these information procedures. A component "c*onditions of simulation"* contains a set of another program of "information procedures" needing for data collection and processing of data during simulation run.

Conditions of simulation and information of procedures present an "algorithm of investigation".

Simulation system Triad includes a library of standard information procedures but an investigator may describe an information procedure (and conditions of simulation too) with the help of Triad language:

information procedure<*name*>*(<a list of generic parameters>)(<input and output formal parameters>)*

initial <*a sequence of statements*> ***endi*** <*a sequence of statements*> ***processing*** <*a sequence of statements*>*…**endinf*** {it is the description of information procedure}.

The linguistic construction of **c*onditions of simulation:***

Conditions of simulation <*name*>*(<a list of generic parameters*>)(<input and output formal parameters>) **initial** <a sequence of statements> ***endi*** <a list of information procedures> <a sequence of statements> ***processing*** <a sequence of statements>…***endcond***

Simulation experiment

The investigation of the structure layer only is static process. The simulation process may take place only after the definition of the behavior of all nodes of model in structure layer. As it was noted above the behavior is determined by the statement ***Put***. It is well known that a simulation is a set of object functioning according to some definite scenarios controlled by synchronizing algorithm. The simulation run is initialized by the statement ***simulate***:

Simulate <a list of models> ***on condition of simulation*** <имя условия моделирования>(<настроечные параметры>)(<параметры интерфейса>)(<список информационных процедур>; <последовательность операторов>).

One can pay an attention to the fact that the several models may be simulated under the same conditions of simulation simultaneously.

Moreover it is possible to define several parts of linguistic construction "*conditions of simulation*" in the statement *simulate*. It is rather important. Let us suppose that an investigator wish to design computer network, so it is necessary to define the structure of computer network and characteristics of workstations and other devices of computer network. Let us suppose that an investigator want to define the configuration of computer network and technical characteristics of cheapest network, of more efficient network, of more secure one and etc. So it is necessary to define different criteria during simulation run and one can do it using different "conditions of simulation" in statement simulate.

Ontological approach to multi model simulation

It is important to involve into the simulation process not only the specialists in simulation but the specialist in specific domains and specialists in the other spheres of knowledge. That is why it is necessary to adjust a simulation system to specific domain. Indeed the investigator of computer network may use a graph theory while studying the structure of network, or a queue network theory, or the theory of Petri Nets. Ontologies are used in TriadNS to set the simulation system to specific domain. Let us briefly consider the decision of this problem in TriadNS.

Basic ontology

Ontologies can be applied on the different stages of simulation. Very often ontologies are applied for the simulation model assembly [Liang & Pardis, 2003]. So the simulation model may consist of separately designed and reusable components. These components may be kept in repositories or may be found via Internet. The ontologies keep the information about interconnections of simulation model components and other characteristics of these components.

It is well known that ontologies enable investigators to use one and the same terminology, so it is one more argument for the use of ontologies. The basic ontology is designed in TriadNS[Mikov et al, 2007].

Its basic classes are: TriadEntity (any named logic entity), Model (simulation model), ModelElement (a part of simulation model and all the specific characteristics of a node of a structure layer), Routine (node behavior), Message (note, please, that nodes of simulation model can send message to any other message) and so on.

The basic properties of the basic ontology in TriadNS are:

– *The property of ownership:* model has a structure, a structure has a node, a node has a pole and so on;

– *The property to belong to something* — inverse properties to previous one– The structure belongs to the model, the node belongs to structure, the pole belongs to the node and so on;
– *The properties of a pole and an arc connection:* connectsWithArc (Pole, Arc), connectsWithPole (Arc, Pole);
– *The property of a node and an appropriate routine binding*-putsOn (Routine, Node);
– *The properties of a node and an appropriate structure binding:* explicatesNode (Structure, Node), explicatedByStructure (Node, Structure);
– *The property of the model and conditions of simulation binding* (Model, ModelingCondition).

The fragment of basic ontology is present below (see Figure1):

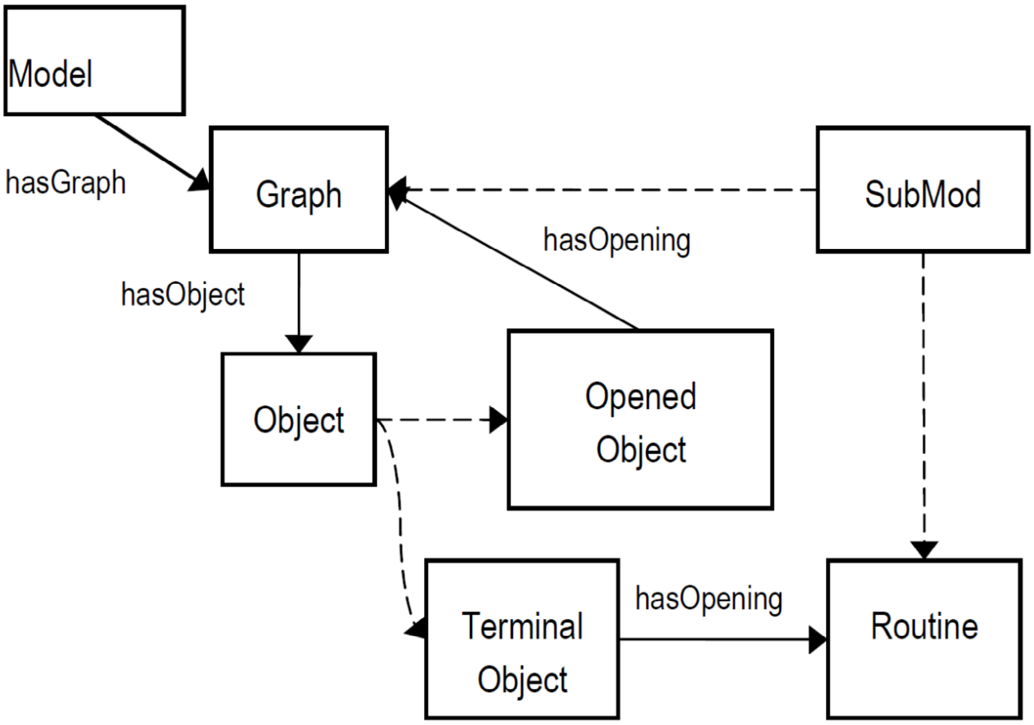


Figure 1. Basic ontology in TriadNS

The basic ontology is presented by the language OWL in simulation system TriadNS. OWL was chosen because there is a big number of programming tools for OWL processing, for the publication of ontologies in the Internet, for searching information (maybe appropriate ontologies) via Internet. Authors use Jena OWL API.

Program tools TriadNS uses ontologies in several cases: (a) for the completeness of (as it mentioned above, simulation model has a structure layer (a several of nodes and edges connecting these nodes) and a routine layer (every node has a behavior defining by routine. One may name simulation model as incomplete or partly defined if any node haven't pointer to a routine defining it. Let us name such a node as "incomplete node"); (b) for integrating simulation models; (c) for construction simulation model in accordance to a specific domain (the main subject of this paper).

Let us describe the process of the completeness of partly defined simulation model in TriadNS more precisely:

- Special subsystem TriadBuilder [Mikov & Zamyatina, 2010] attempts to search the appropriate routine by the help of base ontology (it was described earlier). It may be found thanks to special semantic type (semantic type "Router" and "Host", for example);
- Model completion subsystem starts when the internal form of simulation model is not built according to a Triad code (it has incomplete nodes);
- First, module "model analyzer" searches all incomplete nodes of a simulation model, and marks them. After the *inference* module starts looking for an appropriate routine instance for each of marked nodes according to specification condition (the semantic type of node and routine must coincide).Then the condition of configuration must be checked (the number of input and output poles of node and the number of poles of routine must coincide). Third condition – the poles of incomplete nodes must be connected with the poles of appropriate nodes;
- So the simulation model can be completed by appropriate routines.

Ontology for a simulation of computer networks

The simulator TriadNS has some additional special subclasses of the base classes (specific domain – computer networks) [Zamyatina et al, 2011]:

- *ComputerNetworkModel* (a model of a computer network*), ComputerNetworkStructure* (a structure of a computer network model)*;*
- *ComputerNetworkNode* (a computer network element, it contains several subclasses: Workstation, Server, Router);
- *ComputerNetworkRoutine* (a routine of a computer network) and so on.

This ontology includes two special properties of a pole. These properties are used to *check* the conditions of matching routine to a node:

- *isRequired(ComputerNetworkRoutinePole, Boolean)* – this property check if it is necessary to connect a pole with another pole?
- *canConnectedWith(ComputerNetworkRoutinePole, ComputerNetworkRoutine)* – this property check the semantic type of an element of a structure being connected.

Ontology representing the models of computer networks as the queuing network

The ontology of semantic types of the queuing network supplements the basic ontology in the following way. Firstly, subclasses of the basic ontology classes specific for a queuing network domain are included: MpsModel – a model of the queuing network, MpsStructure – structure of the model of the queuing network and so on.

Secondly, four subclasses of the node class which correspond to the basic elements of the queuing network are included: Generator, Queue, Channel, Terminator.

Third, the classes representing the information specific for the queuing network are included: class of arrival time distribution, class of a service time distribution and so on.

Hierarchy of semantic types in ontology is represented as a hierarchy of routine classes and its top element is subclass of Routine class of the basic ontology - MpsRoutine that is a routine of the queuing network.

When constructing the hierarchy of semantic types the main attention is paid to the multiple inheritances.

The Figure 2 shows a fragment of the queuing network.

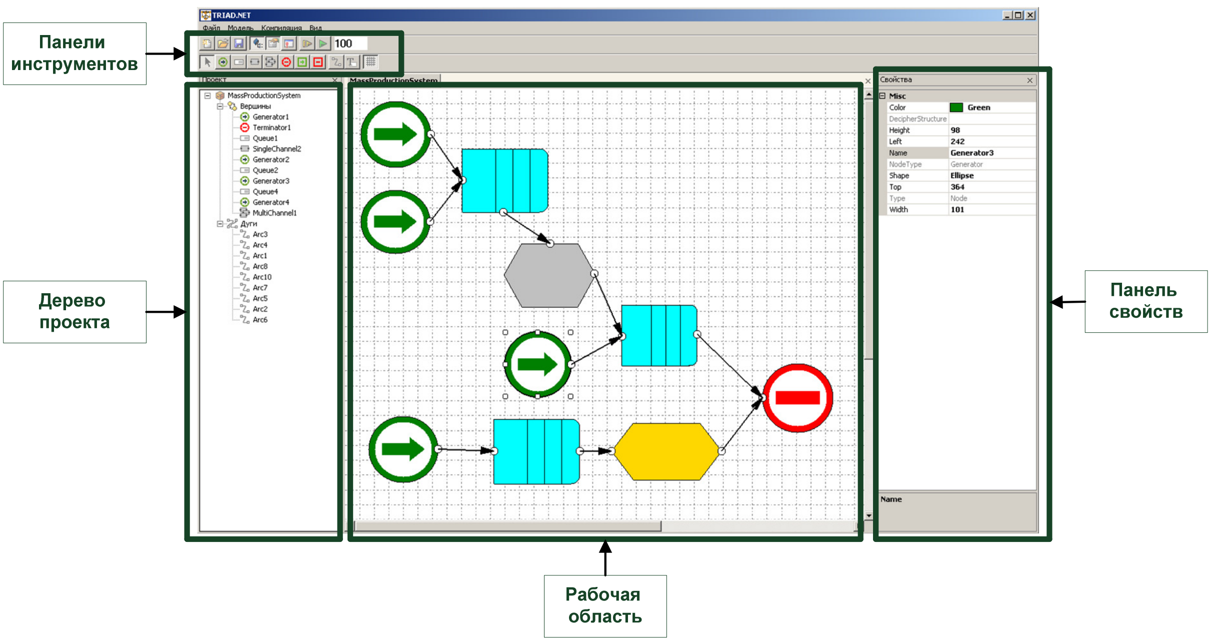


Figure 2. A fragment of queueing network

Conclusion

The paper demonstrates the ontological approach how to adjust the simulation system to a specific domain in the computer network simulator TriadNS.

An adjustment to a specific domain allows improving the quality of the simulation researches because the specialists from the different fields of knowledge can be invited, at that, they can work within their own domains using their standard notions, concepts and terms.

At present the basic and subject-oriented simulation systems are built for computer networks and queueing networks in TriadNS.

The authors of the paper are planning to design program and linguistic tools to transform one model into another (multimodel simulation).

Acknowledgments

The reported study was supported by RFBR, research project No. 13-07-96506_r-s-a "Design of deontic foundation of multiagent technologies of knowledge base construction for virtual laboratories"

Bibliography

[Benjamin et al, 2005] Benjamin P., Akella K.V., Malek K., Fernandes R. "An Ontology-Driven Framework for Process-Oriented Applications"// Proceedings of the 2005 Winter Simulation Conference / M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds., pp. 2355-2363.

[Benjamin et al, 2006] Benjamin P., Patki M., Mayer R. J. "Using Ontologies For Simulation Modeling"// Proceedings of the 2006 Winter Simulation Conference/ L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds., pp.1161-1167

[Gribova & Cherkezshvili, 2010] Грибова В.В., Черкезишвили Н.Н. "Развитие онтологического подхода для автоматизации разработки пользовательских интерфейсов с динамическими данными // Информационные технологии.-2010.-No10.-С. 54-58.

[Liang & Pardis, 2003] Liang V.C, Pardis C.J.J. "A Port Ontology for Automated Model Composition"// Proceedings of the 2003 Winter Simulation Conference / S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., pp. 613-622.

[Mikov & Zamyatina, 2010] Миков А.И., Замятина Е.Б. "Проблемы повышения эффективности и гибкости систем имитационного моделирования", Проблемы информатики, №4(8), Новосибирск, Институт вычислительной математики и математической геофизики СО РАН, 2010, С. 49-64.

[Mikov, 1995] Mikov A.I. "Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems" // Gero J.S. and F.Sudweeks F.(eds), Advances in Formal Design Methods for CAD, Preprints of the IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design, Mexico, Mexico, 1995, pp.105-127.

[Mikov et al, 2007] Mikov A., Zamyatina E., Kubrak E. "Implementation of simulation process under incomplete knowledge using domain ontology" //Proceedings of the 6-th EUROSIM Congress on Modeling and Simulation, University of Ljubljana, Slovenia, 2007, Vol.2, Book of full papers, p.1-7.

[Mikov et al, 2013] Mikov A., Zamyatina E., Kozlov A., Ermakov S. "Some Problems of the Simulation Model Efficiency and Flexibility", Proceedings of "2013 8th EUROSIM Congress on Modelling and Simulation EUROSIM 2013", Cardiff, Wales, United Kingdom, pp. 532-538

[Rathnam & Paredis, 2004] Rathnam T., Paredis C.J.J. "Developung Federation Object Models Using Ontologies"// Proceedings of the 2004 Winter Simulation Conference / R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds., pp. 1054-1062.

[Sargent, 2005] Sargent R.G. "Some Recent Advances In The Process World View"// Proceedings of the 2005 Winter Simulation Conference / M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds., pp. 293-298.

[Sokolov & Yusupov, 2005] Соколов Б.В., Юсупов Р.М. "Концептуальные основы квалиметрии моделей и полимодельных комплексов" // Имитационное моделирование, Теория и практика: сб. докл. 2-й всерос. науч.-практ. конф. ИММОД–2005, Т. 1. СПб: ЦНИИТС, 2005, С. 65–70.

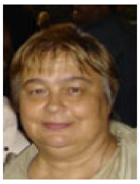
[Сухов, 2013] Сухов А.О. "Инструментальные средства создания визуальных предметно-ориентированных языков моделирования" / Фундаментальные исследования, 2013, № 4 (ч. 4), С. 848-852.

[Zamyatina et al, 2011] Замятина Е.Б., Миков А.И., Михеев Р.А. "Применение онтологий в системе проектирования и моделирования компьютерных сетей TRIADNS", Труды Конгресса по интеллектуальным системам и информационным технологиям, IS&IT'11, Научное издание в 4-х.томах, М. Физматлит, 2011, Т.1, стр. 253-260

[Zamyatina & Mikov, 2012] Замятина Е.Б., Миков А.И. "Программные средства системы имитации Triad.Net для обеспечения ее адаптируемости и открытости", Информатизация и связь, 2012, № 5, С. 130–133.

[Zamyatina et al, 2013] Замятина Е.Б., Лядова Л.Н., Сухов А.О. О подходе к интеграции систем моделирования и информационных систем на основе DSM-платформы MetaLanguage//В кн.: Технологии разработки информационных систем ТРИС-2013. Материалы IV Международной научно-технической конференции/ Т.1.Таганрог: Издательство Технологического института ЮФУ, 2013.с.61-70

Authors' Information



***Elena Zamyatina** – Assoc. Prof. PhD, National Research University Higher School of Economics, Department of Business Informatics; Russia, Perm, 614070, Studencheskaya st., 38;*

e-mail: e_zamyatina@mail.ru

Fields of Scientific Research: simulation, artificial intelligence, ontology, domain specific languages (DSL)

***Alexander Mikov** - Full Professor, Kuban State University, 149, Stavropolskaya str., Krasnodar, Russia, Computing Technologies Department, e-mail: alexander_mikov@mail.ru*

Fields of Scientific Research: information systems, simulation, artificial intelligence, ontology

***Roman Mikheev** - "PROGNOZ", Stakhanovskaya St, 54, Perm, Russan Federation, Programmer, e-mail: miheev@prognoz.ru.*

Fields of Scientific Research: Simulation, Artificial Intelligence, Computer Networks Simulation, Ontologies