Automation of Simulation Steps using Ontological Approach

Elena Zamyatina[1], Alexander Mikov[2] and Viacheslav Lanin[3]
[1]*National Research University Higher School of Economics, Perm, Russian Federation*
[2]*Kuban State University, Krasnodar, Russian Federation*
[3]*Perm State University, Perm, Russian Federation*

Keywords: Simulation, Ontology, Automation of Simulation Steps, Ontological Approach, DSL.

Abstract: There are different Modelling and Simulation (M&S) life cycle's steps described in the literature. One way or another some of these steps are similar: creation of conceptual model, verification and validation of simulation model, statistical data collection and processing. Authors of represented paper suggest to automate some steps and propose to use ontological approach. The automatization of steps allows to optimize the overall time of simulation experiment, to increase a reliability of simulation model and to receive more adequate results.

1 INTRODUCTION

Nowadays simulation became one of most useful and, maybe, single method for investigation of complex systems in various areas of business, economics, healthcare, etc.

The design of simulation models, simulation, analyses of the results of simulation experiment may be fulfilled using the special software tools – simulation systems.

It is well known that simulation process has some steps: a development of conceptual model, verification and validation of model, simulation run, analyses of the results of simulation run and so on (Balci, 1998; Balci et al., 2002; Law and McComas, 2001; Sargent, 2005; Sargent, 2007, Salmon and Aarag, 2011).

Because researcher wishes to receive reliable and truthful recommendation in order to make a decision and because these recommendations have to be received rather quickly it is advisable to optimize steps mentioned above.

Further, we will consider the efforts of the developers of simulation system TriadNS to optimize and automate the steps of the simulation process using the methods of artificial intelligence. One of these methods is ontological approach.

2 MOTIVATION

Simulation usually deals with complex system, thus it is necessary to have sufficient computational resources to shorten the execution time of the simulation experiment, since it is very important that the simulation experiment be completed at a reasonable time (Salmon and Aarag, 2011).

The development of simulation models requires significant efforts from users. Most users are specialists in a specific subject area and do not have the art of programming. For this reason, simulation system is required to have a convenient and user-friendly interface. Moreover it is advisable to have visual programming language describing simulation model.

A useful property of simulation system is the ability to customize the interface to a specific subject area. Users should be able to work in the software environment using familiar terms, operate the construction of the modeling language (including the visual one) (Cetinkaya and Verbraeck, 2011).

The task of the designer is to build the most "adequate" model. "Adequate" model have to describe an investigated object or situation in detail and rather precisely. An "adequate" model can be obtained using a multi-model approach (Sokolov and Uysupov, 2005), transforming one model into another in the course of research.

As a result of the simulation experiment, the user often receives a large amount of unstructured data.

Zamyatina, E., Mikov, A. and Lanin, V.
Automation of Simulation Steps using Ontological Approach.
DOI: 10.5220/0006933102230230
In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018) - Volume 2: KEOD*, pages 223-230
ISBN: 978-989-758-330-8

Therefore, it is advisable to provide the simulation system with software tools for additional data processing so that additional conclusions can be drawn about the results of the simulation. It is necessary to make effective recommendations and to receive the most appropriate decision.

So simulation system must have:

1. Flexible software and language tools for simulation model development.
2. Software tools and languages (may be visual) for optimizing the simulation experiment in time.
3. Software tools for verification and validation of simulation model.
4. Software tools and languages for data collection, processing and additional tools for analyses (may be with the help of special methods - data mining for example).

Special mechanisms and special data structures are needed to develop flexible software, the purpose of which involves automating and optimizing the steps of a simulation experiment. The authors of this paper attempted to use ontologies.

The paper will then be structured as follows: simulation model in the TRIADNS modeling system, basic ontology, ontologies automating the creation of a simulation model, ontologies for customization on a specific subject area, simulation model transformation.

Let us consider several papers discussing how ontologies may be used in simulation systems and how ontological approach may be applied for simulation steps automation and optimization.

3 RELATED WORKS

It is well-known that ontology is defined as a method of representing items of knowledge (ideas, things, facts, etc.) in such a way that determines the relationships and classifications of the concepts within a specified domain of knowledge (Zaletelja, 2018).

Ontologies allow researchers, domain experts, and software agents to share a common understanding of the concepts and relationships of a domain. So ontologies are used in several simulation systems for a large number of domains (a lot of publications about application ontologies in different domains appeared).

Thus, a foundational ontology for manufacturing – system modelling is proposed in (Zaletelja, 2018) Quoting the authors, we can say that “by the formal definitions of the modelling environment itself enable the definition of the manufacturing system’s elements”. Ontological-based approach “ensures the consistency of ever-changing models”.

Modeling framework based on an ontology network is described in (Sarli, 2005). This framework is created to conceptualize supply chain (SC) and simulation domains, besides through the execution of derived axioms, integrity axioms and rules in the ontology network the composition of the SC model is validated.

A methodology and applications of ontology-based simulation are presented in (Beck et al., 2010). An environment for building simulations based on the Lyra ontology management system is described. This system includes web-based visual design tools for constructing models and automatically generating simulation code.

Ontological-driven simulation systems were discussed in (Benjamin et al., 2005; Benjamin et al., 2006). The key motivations of proposed investigations are: to allow for the decomposition of the target system into smaller, to distribute the model development effort among different functional groups and then assemble a simulation model of the entire target system. Authors precisely discussed the problem of simulation model interoperability emphasizing syntactic interoperability and semantic one. Syntactic interoperability deals with the interoperability of implementation details (parameter passing mechanisms, external data accesses, timing mechanisms, for example). Semantic interoperability “deals with the validity and usefulness of translated/composed simulation models”. Authors in (Benjamin et al., 2006) described a structured ontology-driven methodology for interoperability of simulation models.

The use of ontologies to fulfill the development of simulation models encoding knowledge from ontologies is presented in (Silver et al., 2007). This paper discusses a technique that establishes relationships between domain ontologies and a modeling ontology and then uses the relationships to instantiate a simulation model as ontology instances. Techniques for translating these instances into XML based markup languages and into executable models for various software packages are also presented.

The use of OWL for representing object-oriented descriptions to support distributed representations of data, behaviors, descriptions of units and objects to be simulated, and scenarios with initial conditions is described in (Lacy et al., 2004). (Fishwick and Miller, 2004) provides an overview of the application of semantic web technology to Modeling and Simulation.

Let us consider now simulation system TriadNS and a representation of simulation model in this system. Later authors will discuss software tools for simulation model development.

4 SIMULATION SYSTEM TriadNS

Simulator of computer networks TriadNS was designed on the foundation of CAD (Computer Aided Design) system Triad (Zamyatina et al., 2012; Zamyatina and Mikov, 2012) in Perm State National Research University. Software system Triad and special language Triad (Mikov, 1995) were devoted to the computer systems design and simulation.

The design and implementation of CAD Triad was renewed in 2002. It was new version of Triad – Triad.Net. New version is written in C#. Several years later special version TriadNS for computer networks design and analyses was implemented.

Simulation model in TriadNS is represented by several objects functioning according to some scenario and interacting with one another by sending messages.

In order to build simulation model in TriadNS it is necessary to define a structure of modeling system (a set of objects which are connected from one to another), the behavior of these objects (a specific scenario). Moreover it is necessary to determine the structure of the data exchanged between objects.

So, all objects in TriadNS may be divided into three parts named “layers” (because we have a hierarchy). These layers are: a layer of structures, a layer of routines and a layer of messages appropriately. And it is necessary to highlight that simulation model is hierarchical one: each object belonged to a layer of structure may be represented as a structure of objects which belong to the lower layer of structure and so on. The layer of routines includes scenarios of a behavior for each object. It is important to outline that each of objects in layer of structure must have a scenario of behavior. A layer of structure is convenient to present by a graph, more precisely 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, a set of external poles serves to send messages to the objects situated on higher or underlying levels of description.

Let us consider structure layer in TriadNS more precisely: structure layer may be described by the linguistic constructions of special language Triad. Moreover, investigator may use graphical editor and so may describe simulation model using visual language. The example of the simulation model of SDN (software defined) network one can see below.

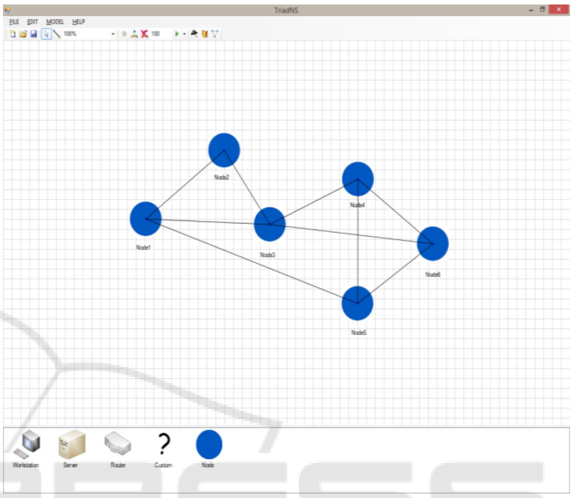

Figure 1: Simulation model of SDN-network is presented as graph, nodes of graph – computer nodes of network, computer communication lines are presented by edges of graph.

Simulation language Triad has several specific features – graph constants, semantic types and etc. Thus simulation model may be described with the help of graph constants (with parameters): cycle(5) (5 nodes connected by edges), rectan, tree and etc. Graph constants present the basic types of topologies of computer network.

Besides, one may choose semantic type (Router, Host, for example). The semantic types are used for simulation model redefining. More details will be given later.

An investigator may take the description of the node’s behavior in repository (or via Internet) or may describe it using special statements and linguistic construction of Triad-language.

5 BASIC ONTOLOGY IN TriadNS

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 a 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 adjust the simulation system to a specific domain.

Ontologies can be applied on the different steps of simulation process (Benjamin et al., 2005; Benjamin et al., 2006). Very often ontologies are applied for the simulation model assembly. 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.

Moreover, ontologies enable investigators to use one and the same terminology and etc. The basic ontology is designed in TriadNS (fig.2.).

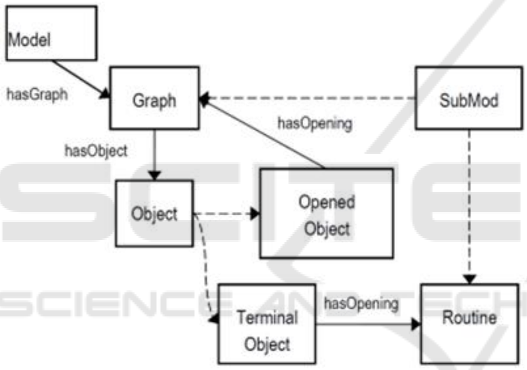

Figure 2: The basic ontology in TriadNS.

It’s basic classes are: TriadEntity (any named logic entites), Model (simulation model), ModelElement (a part of simulation model and all the specific characteristics of a node of structure layer), Routine (node behavior), Message (note, please, that structure layer nodes of simulation model can interchange with messages) and so on.

The basic properties of base ontology are:

1. A property of ownership: a model has a structure, a structure has a node, a node has a pole and etc.
2. A property to belong to something: a structure belongs to the model, a node belongs to a structure, a pole belongs to a node and etc.
3. The properties of a pole and an arc connection: (a) connectsWithArc (Pole, Arc), (b) connectsWithPole (Arc,Pole).
4. A property of a node and an appropriate routine: binding_puts On (Routine, Node).
5. The properties of a node and an appropriate structure: (a) explicatesNode (Structure, Node), (b) explicatedByStructure (Node, Structure).
6. A property of a model and conditions of simulation: binding (Model, ModelingCondition).

The hierarchy of classes of basic ontology is presented below (fig.3.).

6 AUTOMATION OF SIMULATION MODEL DEVELOPMENT

An ordinary simulation system is able to perform a simulation run for a completely described model only. We outlined this fact discussing the layer of

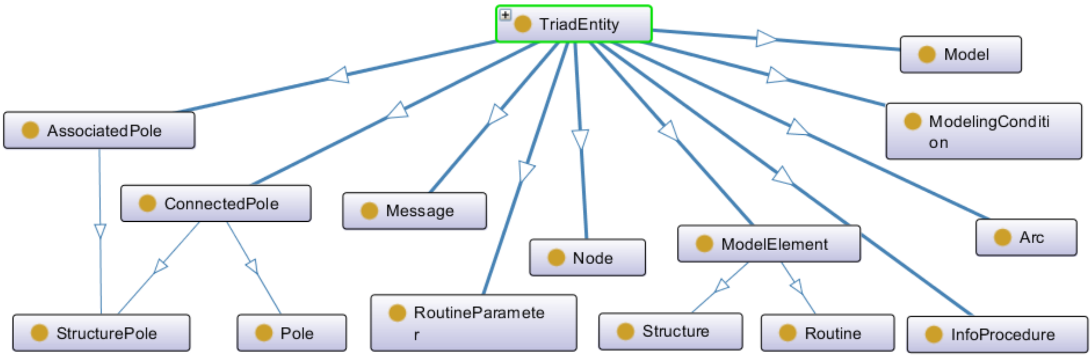

Figure 3: A hierarchy of the classes of the basic ontology in TriadNS.

structure. However, at the initial stage of designing process an investigator may describe a model only *partly* omitting description of a behaviour of one or several nodes (routine of terminal nodes).

Moreover, simulation model may be described without any indication on the information flows effecting the model or without the rules of signal transformation in the layer of messages.

However for the simulation run and the following analysis of the model all these elements have to be described, but this description may be inaccurate and how inaccurate it may be we will describe below.

For example, in a completely described model each terminal node has an elementary routine. An elementary routine is represented by a procedure. This procedure has to be called if one of poles of node receives a message. But some of the terminal nodes of *partly* described model do not have any routines.

Therefore the task of an automatic completion of a simulation model consists either in “calculation” of appropriate elementary routines for these nodes.

It was mentioned above that the routine specifies behavioural function assigned to the node, but the structure graph specifies additional structure level of the model description. And at the same time, all structures must be completely described as the sub models.

These actions have to be fulfilled by the special software tool TriadBuilder (one of the components of simulation system TriadNS).

Software tool TriadBuilder attempts to search the appropriate routine by the help of basic 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 built according to a Triad code.

First, model analyser searches the model for incomplete nodes, and marks them. Thus, the model analyser will mark all nodes without routines.

Then 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). An appropriate routine may be found in repository or in Internet.

If it is not opportunity to find appropriate routine by a semantic type another conditions must be checked.

It is a condition of configuration: the number of input and output poles of node and the number of poles of routine must coincide.

After the appropriate instance of routine has been found, it may be put on the node.

If the condition of configuration was not met, then the new last condition must be checked: it is a check of the “environment” graph. An environment graph is a graph of nodes connected to a node marked with component TriadBuilder. Rather, the semantic type of nodes of this graph.

7 AUTOMATION OF SIMULATION MODEL DEVELOPMENT FOR A SPECIFIC DOMAIN

The simulator TriadNS has some additional special subclasses of the basic classes (specific domain here– computer networks):

1. ComputerNetworkModel (a model of a computer network),
2. ComputerNetworkStructure (a structure of a computer network model),
3. ComputerNetworkNode (a computer network element, it contain several subclasses: Workstation, Server, Router),
4. ComputerNetworkRoutine (a routine of a computer and etc.

Moreover this ontology includes two special properties of a pole.

These properties are used to check the conditions of matching routine to a node:

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

Nowadays SDN (software defined networks) and SON (self - organizing networks) become popular. These networks are the attempt to simplify and speed up the planning, configuration, management, optimization of communications networks.

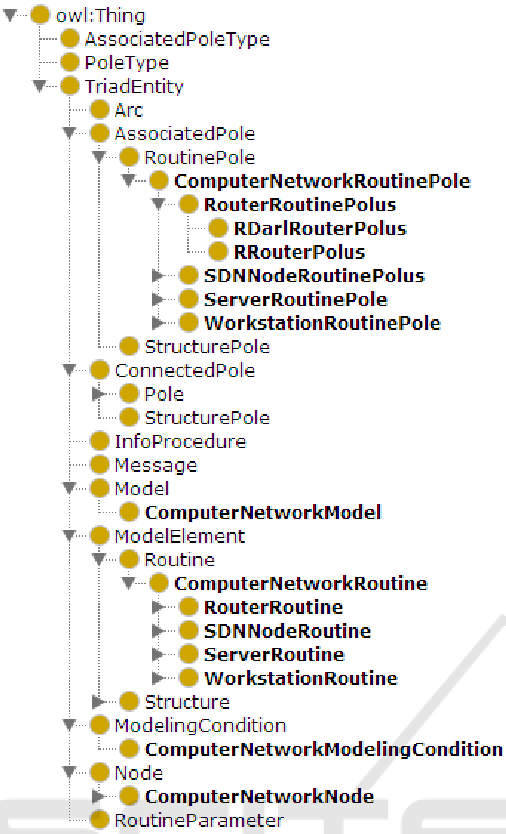

Figure 4: A hierarchy of the classes of the ontology including classes and subclasses describing computer networks in TriadNS.

The investigation of the routing algorithm SBARC for SDN were carried out.

Thus it was necessary to design several new subclasses:

1. subclass (SDNNode) of class ComputerNetworkNode,
2. subclass SBARCRoutine,
3. subclass SDNNodeRoutine of class ComputerNetworkRoutine.

One can see a hierarchy of the ontology classes describing computer networks in TriadNS, in fig.4.

So one can see that including new classes and subclasses to basic ontology allows to expand the possibilities of the simulation system. Moreover, the model is validated and it is carried out automatically.

Let us consider another example: transformation of conceptual simulation model to a model described with the help of another visual language.

8 SIMULATION MODEL TRANSFORMATION

Usually simulation systems allow investigating objects (or situation) with the help of mathematical scheme – queuing networks. But it is advisable to apply other mathematical schemes, for example, Petri nets or Markov processes.

Special software components of TriadNS simulation system (embedded software tools) allow transforming conceptual model to the model which is described with special visual languages (MDE (model driven engineering approach) (Chetinskaya and Verbraeck, 2011) implemented in TRIADNS. One can see the model of simple computer network consists of two workstations and one server and this transformed model. This model presents as Petri Net.

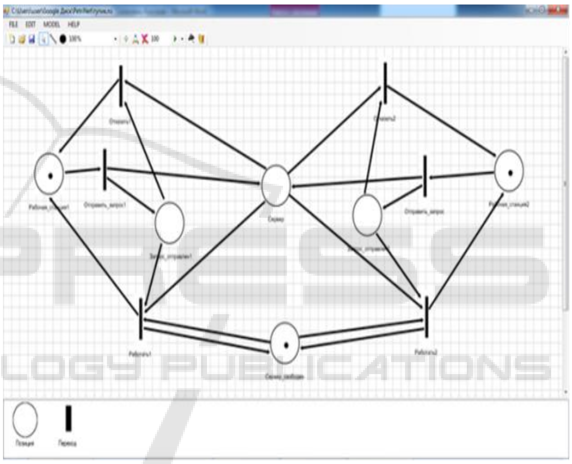

Figure 5: Conceptual model of computer network is transformed to Petri net.

In order to fulfill model transformation the basic ontology was enriched with new classes and subclasses, and the TriadNS software tools – with software that implemented the rules for translating one model to another. The ontology for Petri Net is presented below.

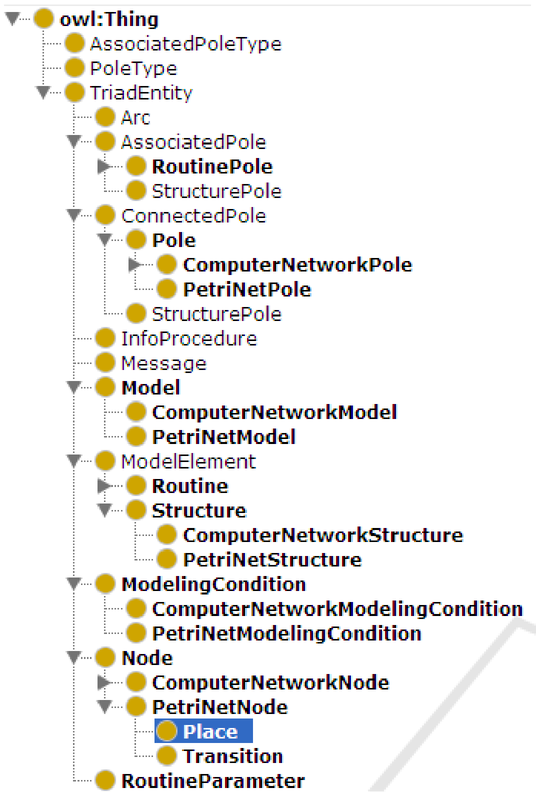

Figure 6: Conceptual model of computer network is transformed to Petri net.

9 CONCLUSIONS

So we considered ontological approach using by the developers of simulation system.

Ontological approach was applied for automation of several steps of simulation process: redefining of simulation model, adjusting of conceptual simulation model to a specific domain (embedded DSL component). Moreover one more very important step of simulation process was implemented in TriadNS: verification and validation. Special software component build ontology of errors. After the step of translation this component carries out the elimination of errors due to rules specified in ontology.

Thus the automation of these steps allows investigators to carry out simulation in convenient software environment, to involve in simulation process specialists in various specific domains and in this way to receive more adequate simulation models and appropriate results. Moreover ontological approach help to reduce the overall time needed for simulation.

ACKNOWLEDGMENTS

The reported study was funded by RFBR according to the research project № 18-01-00359.

REFERENCES

Balci, O., 1998. Verification, Validation, and Accreditation. *Proceedings of the 1998 Winter Simulation Conference. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds., Grand Hyatt Washington, Washington DC, pp.41-48.*

Balci, O., Nance, R. E., Arthur J. D., 2002. Expanding Our Horizons In *Verification, Validation And Accreditation Research and Practice Proceedings of the 2002 Winter Simulation Conference E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds., WSC 2002, San Diego, California, pp.653-663.*

Law, A. M., McComas, M. G., 2001. How to build valid and credible simulation models. In *Proc. 2001 Winter Simulation Conf., ed. B.A. Peters, J. S. Smith, D. J Medeiros, and M. W. Rohrer, Piscataway, New Jersey: IEEE. pp. 22-29.*

Sargent, R. G., 2005. Verification and validation of simulation models. Proceedings of Winter Simulation *Conf., ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, Piscataway, New Jersey, 2005: IEEE pp.130-143.*

Sargent, R. G. 2007. Verification and Validation of Simulation Models. *Proceedings of Winter Simulation Conf., S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds., J.W. Marriott Hotel, Washington, 2007, pp.124-137*

Salmon, S., Aarag, H., 2011. Simulation Based Experiments Using Ednas: The Event-Driven Network Architecture Simulator. In *Proceedings of the 2011 Winter Simulation Conference S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.. The 2011 Winter Simulation Conference 11-14 December 2011 Grand Arizona Resort Phoenix, AZ, pp. 3266-3277*

Cetinkaya, D., Verbraeck A., 2011. Metamodeling and model transformations in modeling and simulation // *Proceedings of the 2011 Winter Simulation Conference, 2011, pp. 3048-3058*

Sokolov, B. V., Uysupov, R. M. 2005. Konceptualnye osnovy kvalimetrii modeley i polimodelnyh kompleksov//Imitacionnoe modelirovanie. Teoria I praktika. Sbornik dokladov vtoroi vserossiyskoy nauchno-prakticheskoi konferencii IMMOD-2005. Tom 1. Spb..: CNIITS. 2005.-s.65-70.

Zamyatina, E. B., Mikov, A. I., Mikheev, R. A., 2012. Lingvisticheskie i intellektualnye instrumentalnye sredstva simulyatora kompiutornyh setei TRIADNS. *International Journal "Information theories & Applications (IJ ITA). Vol 19, Number 4, 2012, pp.355-368. ITHEA, Sofia, 1000, P.O.B. 775, Bulgaria.*

Zamyatina, E. B., Mikov, A. I., 2012. Programmnye sredstva imitatsii Triad.Net dlya obespecheniya eyo adaptiruemosti I otkrytosti. Informatizatsiya i svyaz. №5, 2012, ANO «Redaktsiya zhurnala «Informatizatsiya I svyaz», ISSN 2078-8320, PP.130-133[in Russian].

Mikov, A. I., 1995. 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*.

Benjamin, P., Akella, K. V., Malek, K., Fernandes, R. 2005. 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, P., Patki, M., Mayer, R. G. 2006. 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

Mikov, A., Zamiatina, E., 2014. Program Tools and Language for Networks Simulation and Analysis. Proceedings SDN & NFV – The Next Generation of Computational Infrastructure: *2014 International Science and Technology Conference «Modern Networking Technologies (MoNeTec)» October 27-29, 2014 Lomonosov Moscow State University, pp. 94-102*

Mikov, A., Zamyatina, E., Kubrak, E., 2007. Implementation of simulation process under incomplete knowledge using domain ontology. In *proceedings of 6-th EUROSIM Congress on modeling and Simulation. 9-14, September, 2007, Ljubljana, Slovenia, Vol.2. Full papers, 7 p.*

Sarli, J. L., 2005. Ontology-Based Modeling Framework to Generate Federation Object Model in the Supply Chain Domain. Proceedings of Winter Simulation Conf., ed. W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds., *Piscataway, New Jersey, 2005: IEEE pp.4636-4637*.

Beck, H., Morgan, K., Jung, Y., Grunwald, S., Kwonb, H., Wub, J., 2010. Ontology-based simulation in agricultural systems modeling. *Agricultural Systems, 103 (2010) pp.463–477*.

Zaletelja, V., Vrabičb, R., Hozdićb, E., Butalab, P. A., 2018. foundational ontology for the modelling of manufacturing systems. *Advanced Engineering Informatics, 38, 2018, pp. 129–141*

Lacy, L., Gerber, W., 2004. Potential Modeling and Simulation Applications of the Web Ontology Language – OWL. In *Proceedings of the 2004 Winter Simulation Conference, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, pp.265–270*.

Silver, G. A., O. A. Hassan, and J. A. Miller. 2007. From Domain Ontologies to Modeling Ontologies to Executable Simulation Models. In *Proceedings of the 2007 Winter Simulation Conference, eds. S. G. Henderson, B. Biller, M. H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, pp. 1108–1117*.

Fishwick, P. A. and J. A. Miller. 2004. Ontologies for Modeling and Simulation: Issues and Approaches. In *Proceedings of the 2004 Winter Simulation Conference, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, pp. 259–264*.