Simulation Model Validation based on Ontological Engineering Methods

Elena Zamyatina[1][a], Denis Churin[1], Viacheslav Lanin[1][b], Lyudmila Lyadova[1][c] and Nada Matta[2][d]
[1]*Department of Information Technologies in Business, HSE University, Perm, Russian Federation*
[2]*University of Technology of Troyes, Troyes, France*

Keywords: Simulation Model, Verification, Validation, Ontology Matching.

Abstract: A task of the simulation models examination (verification and validation, V&V) is considered. At the V&V process the correspondence degree of the simulation model created by developers to the simulated object, that description is presented in the form of a conceptual model built by customers, is determined. An ontological approach is proposed to determine the semantic proximity of the simulation model and the conceptual model, whose descriptions are presented in the form of ontologies. Matching rules can also be defined with ontology based on the metrics chosen by the customer. The approach has been tested using the simulation system Triads. The results of the matching algorithm execution are illustrated by an example. The article provides description of the simulation model ontology created in TriadNS and conceptual model ontology, developed with MASK method. The metrics used for proximity assessment are described.

1 INTRODUCTION

It is well-known that there are a great number of publications devoted to the examination of the simulation models built by researchers to solve tasks of designing or forecasting, etc. (Sargent, 2017; Sargent, 2007; Balchi, 2004). Examination involves the implementation of such stages as verification and validation (V&V) of a simulation model. It is necessary to prove that this model can be trusted that the quality of information obtained because of a simulation experiment corresponds to the level that allows making the right decision.

Before performing simulation experiments, customers can build a conceptual model, representing the object (system, process) being simulated, as "a reference model". To develop a conceptual model, a suitable environment can be used by customers that does not require programming skills, the use of simulation tools. So, verification comes down to control of the correctness of the transfer of the conceptual model, developed by the researcher, into the simulation environment (Sargent, 2017).

Validation can be defined as examination of the conceptual model presentation correctness and checking behavior correctness. *Verification* is a part of validation. Verification is carried out during the construction of the model, while validation is carried out immediately after the completion of the creation, the description of the model in a specific specialized programming language. However, in practice, as a rule, the processes of verification, validation, as well as testing and implementation of the model overlap in time. Along with the concepts of validation and verification, the literature provides such a concept as accreditation (VV&A – Validation, Verification and Accreditation). *Accreditation* may be defined as the formal certification that a model, simulation, or combination of models and simulations is acceptable for use for a specific purpose (Sargent, 2007). Accreditation is the official certificate of the customer, which states that the simulation model is applicable for solving a concrete problem.

[a] https://orcid.org/0000-0001-8123-5984
[b] https://orcid.org/0000-0002-0650-2314
[c] https://orcid.org/0000-0001-5643-747X
[d] https://orcid.org/0000-0001-8729-3624

Zamyatina, E., Churin, D., Lanin, V., Lyadova, L. and Matta, N.
Simulation Model Validation based on Ontological Engineering Methods.
DOI: 10.5220/0011589000003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 2: KEOD*, pages 237-244
ISBN: 978-989-758-614-9; ISSN: 2184-3228

Verification and validation should answer the question of whether the model is sufficiently accurate. In this case, the purpose of the developed simulation model should be taken into account. For example, most of the demonstration models are not highly accurate, but it is worth remembering that their purpose is to demonstrate the operation of the process, object or system under study only in general terms. If the model copes with the goal set for it, then we can talk about its compliance with the prototype, even though the model's accuracy is low. Therefore, the purpose of the model must be known before validating it. The customer must develop *validation rules that are appropriate for the purpose*.

Thus, the task is to *develop a flexible validation tool* that allows not only to create models, but also to adjust requirements for specific purposes of model examination.

This paper presents approach for developing software tools implementing an *ontological approach to verification and validation of simulation model*. This software tools includes portal for the customers and modelers interaction. Customers must submit a conceptual model ontology ($O1$) and simulation model developers also submit their version of the conceptual model ontology ($O2$). Using an algorithm for determining the proximity of ontologies, a comparison of two ontologies is performed and inconsistencies in them are revealed.

System TriadNS is chosen as the simulation tools for performing experiments. This system uses ontology to customize simulation models to specifics of domains. It is proposed to extend the purpose of the ontology for solving model validation tasks.

2 RELATED WORKS

It should be noted that there are different forms of validation:

- validation of the conceptual model (checking the degree of detail of the model);
- data validation (checking the accuracy of the data);
- white-box validation (detailed (micro) validation of the model, determining the accuracy of the model components);
- black box validation (general (macro) verification of model performance, in which it is determined whether the general model provides a sufficiently accurate representation of the real world).

It is well-known that nowadays many simulation systems use ontologies for simulation model design (Jain, 2016). It allows to integrate simulation models (Benjamin, 2006), to adjust the system to a specific subject area quickly and flexibly, to determine one or another mathematical abstraction of simulation model (Queue theory, Petri Nets and so on).

Using the capabilities of the TriadNS system to use ontologies, it is proposed to implement validation tools based on the scheme (Balchi, 2004) shown in Figure 1.

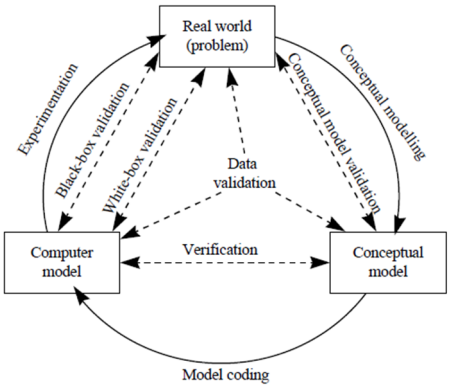

Figure 1: The scheme of verification and validation of simulation model.

Any process can be viewed from different angles and points of view. For example, the process of customer service from the manager's point of view looks like a sequence: call the customer, find out the need, meet the need, complete the service. While the same process for the customer will look like a sequence: choose the right ticket, take a ticket, stand in line, come to the service, voice the need, get service, complete the service. Accordingly, different viewpoints influence what will be considered model validity and how goals are achieved. Different perspectives do not make a model invalid; they only reflect different interpretations of the process. Situations of developing a model from different viewpoints, as in the situation of setting multiple goals, will require a lot of resources because of the high level of detail. Therefore, one model may be valid for one viewpoint and completely invalid for another viewpoint.

The model validation process may involve many tests using data. For example, it may examine what results the model will produce when real data arrive. A model run on the same data as the object of its representation is expected to behave similarly to the real-world object (process or system). In a fairly simple experiment, a number of problems may arise:

1. *Lack of real data*. The experimenter may not have real data to start the simulation process. For example, when simulating queues to ATMs,

information on the average customer flow to bank branches may not be collected. Expert knowledge can be used as a solution to this problem.

2. *Inaccuracy of real-world data*. Due to inaccurate data, model results are not reliable and not suitable for research. That is why data validation is considered as one of the key elements of simulation model building life cycle.

3. *Unrepresentative data sampling*. If the experimenter obtains real world data, such as data from an electronic queuing system, it does not mean that it is representative. For example, the analyst has data on customer flows to ATMs during the holiday season. However, this sample is not representative, and the conclusions are statistically insignificant because the weekday clients flow is very different from the holiday data. Therefore, for current data the model will show results comparable only to the days for which these data are collected. To prevent the problem in question, you should use static tests when validating the data, but the result of the model will have to be seen as the probability of the event occurring. Critical values can be passed to the model to test it and assess the adequacy of the conclusions.

As mentioned before, the process of validation and verification should take place at each stage of model building and should be repeated if the model undergoes changes. Given the complexity of models and the increasing cycle of its construction, validation and verification may require a long period of time, during which the process in the real world may also change. For example, during the validation of the final computer model of the customer service process, in the real-world service standards have changed and the model under study is no longer relevant. To minimize the impact of this problem, the analyst should calculate the time and ensure that the validation and verification as often as possible while evaluating the model as a whole and its elements.

Verification and validation methods are described in more detail in the work of the American researcher Osman Balchi (Balchi, 2004). One is to test the construction of a conceptual simulation model.

The developed conceptual model describes those components of the real-world system that should be included in the model (and those that should be excluded from the final model), and is expressed either formally (for example, using activity cycle diagrams) or informally (for example, in the form list of items).

To create a conceptual model, developers must analyze all the information received and come to an optimal solution. A project specification or terms of reference can be used to validate the conceptual model. In addition, it is necessary to get estimates from experts who are versed in the subject area and similar systems on the compliance of the conceptual model with the requirements described in the documentation.

So, then the stage of verification and validation should be carried out jointly, both by the developers of the model and by the customers who need to solve a specific problem.

As described earlier, the model verification and validation processes affect all stages of the development of a simulation model. The process of checking the adequacy and accuracy of the simulation model includes:

- structural testing (structural validity – determining the correspondence of the structure of the simulation model, the list of objects and their interrelationships to the researcher's intentions);
- testing the functions of the model;
- testing the behavior of the model (operational validity – checking the correspondence of the functionality of the simulation model to the concepts of the researcher).

In addition, simulation model validation should be performed at every stage of simulation model design. If an error is found in the simulation model, it is necessary to return to the previous stage of model verification to identify the discrepancy between the constructed model and the customer's intentions.

A comparison of two ontologies is performed and inconsistencies in them are revealed.

This article focuses on the structural validity only.

Verification is performed by comparing the ontologies representing the conceptual models received from customers and from modelers designing simulation models.

3 SIMULATION MODELING IN TriadNS SYSTEM

TriadNS is a simulation system which was developed based on the simulation system Triad. Triad is intended for the design and simulation of computer systems. Simulation models are described using special language named “Triad”.

It has a three-layer representation of the simulation model: $M = (STR, ROUT, MES)$, where *STR* is the structure layer, *ROUT* is the routine layer, *MES* is the message layer.

A layer of *structures* is a collection of objects interacting with each other by sending messages.

Each object has poles (input and output poles), which serve, respectively, for receiving and transmitting messages. The basis of the representation of the layer of structures is a *graph*. Separate objects should be considered as the *nodes* of a graph. The *arcs* of the graph define the relationships between objects. The structure layer is described as a *parameterized procedure* and allows to flexibly change the number of nodes in the graph, etc. One may change the input parameters both before the start and during the simulation experiment. In the second case, the override is performed in the special unit of simulation model description: "*the conditions of simulation"*.

Model objects act according to a specific scenario, which is described by "*routine*". The state of the object is determined by the values of the variables of the routine. The simulation system TriadNS is *event-driven*. *Events* are described in routines. A routine, like an object, has input and output poles. The input poles are used for receiving messages, and the output poles serve for transferring messages. Message receiving is an event.

The *message layer* is intended to describe messages of complex structure (it may be a program, for example).

The collecting information on the simulation experiment is carried out with *information procedures*. Information procedures work like sensors and monitor the changes in the values of variables, the arrival and sending of messages, and the execution of events. The list of information procedures is specified in a special program unit – the *conditions of simulation*. The conditions of simulation determine the initial conditions during the simulation experiment, the moment of completion of the simulation and determine the algorithm for the study of the simulation model. These tools can be used to validate model checking in simulation experiments, in dynamics. But in this paper, the focus is only on static structural characteristics.

4 TriadNS SIMULATION SYSTEM ARCHITECTURE

Modelers can use different tools to develop models in the TriadNS system. They can create models with *text editor* using Triad language to describe structure, routines, and other elements of the model. Modelers can use a *graphical editor* to create visual models of the simulated systems (the developed models are transformed into a textual representation in the Triad language and translated into executable code to perform simulation experiments). Created models and components of models (structures, routines, information procedures, modeling conditions) can be stored in a *model repository* and reused for new experiments. The generalized structure of the TriadNS system repository is shown in Figure 2.

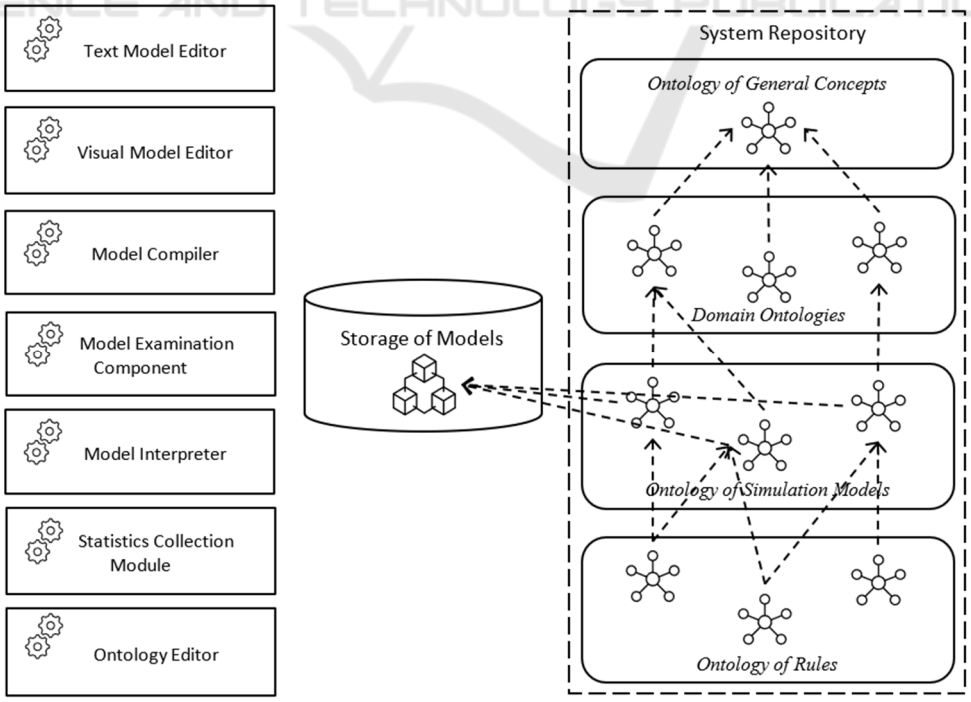

Figure 2: The generalized structure of the TriadNS system.

The *ontology* is the core of the system repository (Zamyatina, 2018). The ontological representation of the simulation model has a multilevel structure. The upper level (the level of *general concepts*) is an ontology that describes general concepts for various mechanisms of the simulation system. The second level of the ontology is the level of concepts represented in *domain ontologies*. Ontologies of this level include concepts specific for systems and processes related to different domains. This level allows to set the semantics of the models, facilitate their understanding by experts in the relevant research areas, and the interpretation of the results of simulation experiments. The third level is the level of *simulation models*. Models and their components are described at this level. Model representations can be associated with specific areas described by domain ontologies. Model components are associated with domain concepts. The lower level of the system ontology (*rule level*) contains descriptions of specific rules that supplement the descriptions available at the previous level. This layer specifies the design rules, the basic requirements that the simulation model must meet.

Ontologies of conceptual models created by customers are used to solve simulation models examination tasks. The rules that are used for model comparison are also be represented in the ontology as descriptions that can be defined for specific modeling projects. The ontology of a conceptual model can be built automatically based on the model created by developers with using a visual model editor or with a text model editor (Figure 3). The customer's conceptual model ontology (Figure 4) can be created with an ontology editor.

5 ONTOLOGY CREATION METHOD

The main problem of applying the proposed model examination method is the development of an ontology by customers who do not have the skills of knowledge engineers.

To translate the requirements and logic of the conceptual model into a machine-readable form of ontology, it is necessary to resort to knowledge extraction tools (Matta, 2002). At the moment, two approaches to knowledge extraction are widely known:

- REX is a method of direct knowledge extraction using Data Mining and Text Mining tools from regulations and documents.
- MASK is a knowledge gathering method based on interviewing experts.

One of the significant advantages of the MASK method over a similar one is a significant saving of time when interviewing an expert, analyzing, and modeling, due to joint work with an expert and using his experience in a specific subject area.

Within the framework of this work, to build the ontology of the conceptual model, the MASK method is mainly used to accurately determine the requirements and vision of the network model by an expert. MASK knowledge gathering helps the expert to focus on his subject area to describe it by emphasizing its main characteristics.

The list of questions is determined by the specifics of the modeling domain. In this case when interviewing an expert to create a conceptual model ontology, he was asked the following questions:

1. What network elements are used to build the model?
2. How many network elements of each type are used in the model?
3. What are the characteristics of the components of a model?
4. What are the relationships between the components of the model?
5. What is the routing algorithm for the model nodes?
6. What are the parameters of the routing algorithm in this model?
7. What are the conditions of simulation experiment?
8. What is the limit time of simulation?

Depending on the answers of experts, the list of questions is detailed, the answers are specified. For example, for the of the routing algorithm the following parameters were obtained:

- two parameters (ST11 and ST12) affect the message processing time (T1+Random(T2));
- two parameters (ST21 and ST22) affect the frequency of sending messages to nodes;
- parameter SQueueLen defines the buffer size;
- parameter SBBCon defines whether there is a broadband connection;
- parameter STFlops specifies the performance or computing power of the node, measured in flops (the number of floating-point operations per second);
- and so on.

One may see these parameters at Figure 4 where the conceptual model ontology of a customer is shown (the basic ontology enriched by subclasses for SBARC algorithm).

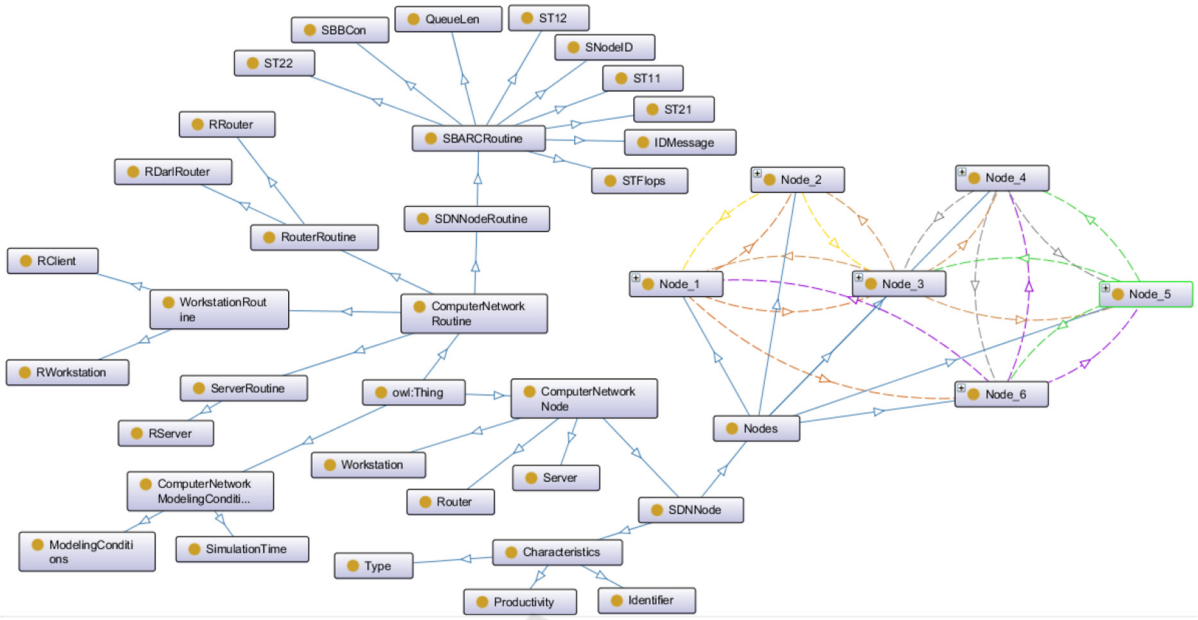

Figure 3: The basic ontology enriched by subclasses for SBARC algorithm (the conceptual model ontology of a developer).

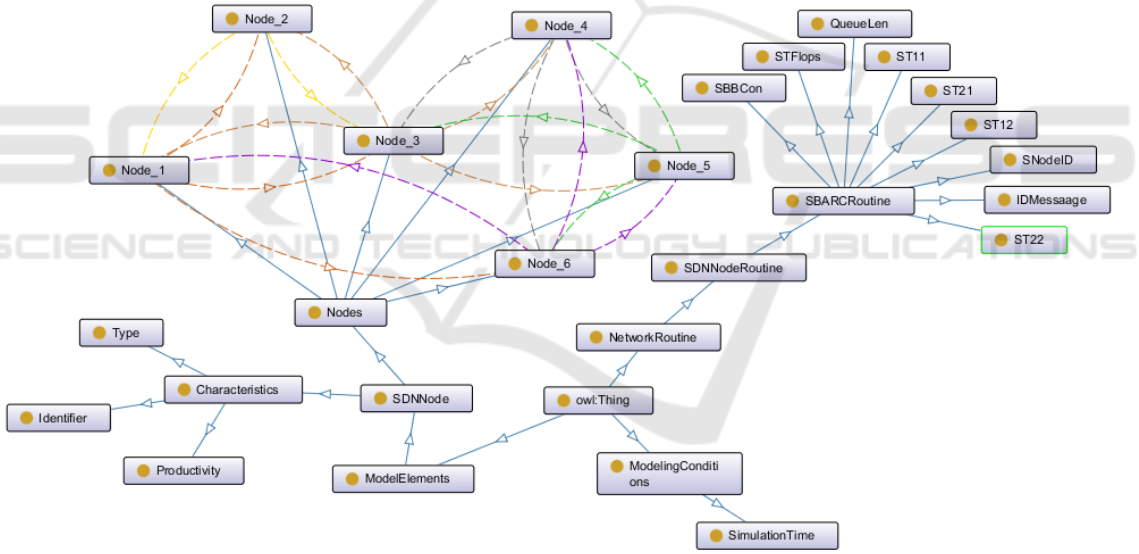

Figure 4: The basic ontology enriched by subclasses for SBARC algorithm (the conceptual model ontology of a customer).

6 STRUCTURAL VALIDATION IN TRIADNS SYSTEM

The ontological approach is actively used in the TriadNS simulation system (Zamyatina, 2018). Ontologies are used to customize model to the domain specifics. TriadNS system has a *basic ontology* consisting of basic classes *TriadEntity* (any named logic entities), *Model* (simulation model), *ModelElement* (a part of simulation model and all the specific characteristics of a node of structure layer), *Routine* (node behavior), *Message* and so on.

Let us consider, as an example, illustrating the possibilities of the ontological approach to model checking, the SDN network (Software-Defined Network). A data transferring control level is separated from the data transferring devices, and it is implemented in software (Zamyatina, 2017).

The basic ontology of the TriadNS simulation system was extended to model SDN. The modelers

got the opportunity to operate with such objects as a router, workstation, super-node, node, router etc.

For the experiment, the SBARC algorithm is simulated (Zhiyong, 2008). Using the graphical editor of the TriadNS system, a network structure was created, including six nodes and connections between them. To implement the SBARC algorithm in model, it is necessary to add new elements into the base ontology. A subclass SDNNode is to be added to class "Nodes". All elements of the network simulation model (nodes and connections) are described in the ontology O_1 (Figure 3). New class is created for each node of the network simulation model (Node_1, Node_2, Node_3, Node_4, Node_5, Node_6). This was done due to the fact that each node in the network has its own unique set of attributes and links to other nodes. These relationships are described as properties of objects. The onto-graph for the conceptual model created by customer (O_2) is shown in Figure 4.

The final stage of the research is the comparison (determination of the proximity measure) of the ontologies built by the developer and the customer (checking the conformity of the logic of the conceptual model). The model built by the developer is valid if it meets the expectations and requirements of the expert set out in the conceptual model. The differences in the structure of ontologies would point specifically to the part of the model in which the error was made.

Ontology *comparison rules* are based on the approach described in the paper (Ngom, 2018).

As an example, a comparison method is implemented that includes the following steps:

1. Defining sets of classes: $O_1 \setminus O_2$ – a set of classes represented in the O_1 ontology and not represented in the O_2 ontology; $O_2 \setminus O_1$ – a set of classes represented in the O_2 ontology and not represented in the O_1 ontology; $O_1 \cap O_2$ – a set of classes, presented in both ontologies.
2. Evaluating the semantic proximity measure between the concepts of each set.
3. Increasing the O_1 and O_2 ontologies using the set $O_1 \cap O_2$. At this step, for each class X from $O_1 \cap O_2$ ($X \in O_1$ and $X \in O_2$) a search is made for their descendants in O_1 (or in O_2, respectively), which are added as descendants of the class X into O'_2 (or into O'_1, respectively), if they don't exist in this ontology.
4. Defining set of classes $O'_1 \cap O'_2$, which are a set of common concepts of the ontologies O'_1 and O'_2.
5. Assessment of similarity between ontologies.

The formula

$$N(O_1, O_2) = \frac{\vartheta\overline{SIM}(o'_1)+I_2+\omega\overline{SIM}(o'_2)+I_1}{\vartheta\overline{SIM}(o'_1)+I_2+\omega\overline{SIM}(o'_2)+I_1+\alpha\overline{SIM}(O_1\setminus O_2)+\beta\overline{SIM}(O_2\setminus O_1)}$$

is used to evaluate the ontologies proximity measure. The following designations are used in the formula:

$I_1 = \frac{1}{1+n_2}$ is the ontology O_1 integrity factor (n_2 is the number of O_2 concepts added to extend O_1);

$I_2 = \frac{1}{1+n_1}$ is the ontology O_2 integrity factor (n_1 is the number of O_1 concepts added to extend O_2);

$\overline{SIM}$ is a function of the average value of the ontology concepts similarity degrees determined with the formula:

$$Sim_{(c_1,c_2)} = \frac{2 * depth\ (c_3)}{depth\ (c_1) + depth\ (c_2)},$$

where c_3 is the closest common parent of c_1 and c_2;

ϑ, ω, α and β are parameters that allow to consider the value of proximity in relation to the number of concepts of sets O'_1 and O'_2 and the number of concepts of ontologies O_1 and O_2:

$$\vartheta = \frac{Number\_of\_Concepts(O\prime_1 \cap O\prime_2)}{Number\_of\_Concepts(O\prime_1)+ n_1+n_2},$$

$$\omega = \frac{Number\_of\_Concepts(O\prime_1 \cap O\prime_2)}{Number\_of\_Concepts(O\prime_2)+ n_1+n_2},$$

$$\alpha = \frac{Number\_of\_Concepts(O_1 \setminus O_2)}{Number\_of\_Concepts(O_1)},$$

$$\beta = \frac{Number\_of\_Concepts(O_2 \setminus O_1)}{Number\_of\_Concepts(O_2)}.$$

Inconsistencies identified when determining the semantic proximity of two ontologies are shown in Figure 5. The proximity measure of the network model ontology and the conceptual model ontology according to the described method is 81%.

In an ideal situation, they should be identical, but, in practice, this is practically unattainable due to the possible features of special modelling environments. The conformity equal to 100% is the ideal and desired result when validating the simulation model.

To increase the flexibility of the system, the rules for matching ontologies and determining the integral proximity estimate can be specified by users in the ontology of the system.

7 CONCLUSIONS

The developed method of the simulation model structural validation based on ontologies comparison was tested. The customer view and created simulation model are described with ontologies. The fixed rules for calculating the integral proximity score are used. Experiments showed the practical significance of the proposed approach.

Suggested approach and developed software tools can be useful when developing complex simulation models.

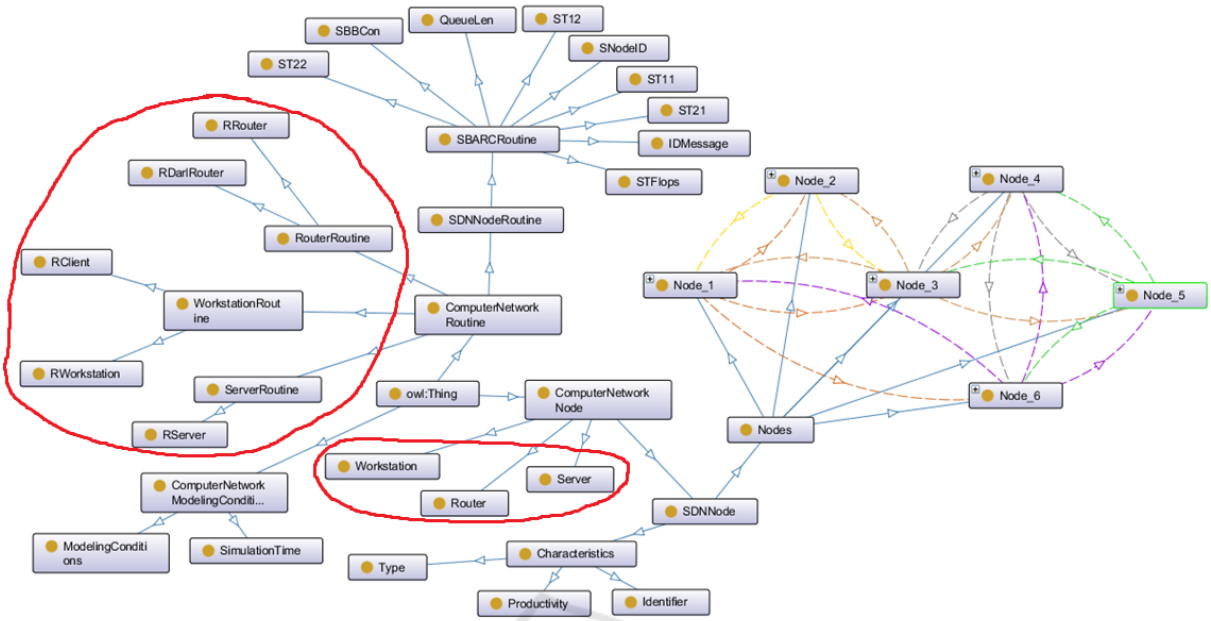

Figure 5: Inconsistencies identified when determining the semantic proximity of two ontologies.

The next step is to increase the flexibility of the approach via providing users with the ability to develop their own metrics, to create rules for comparing models, customizable to specific modeling tasks and domains, with using ontology.

ACKNOWLEDGEMENTS

The reported study was funded by RFBR and the Krasnodar region Administration, project number 19-47-230003.

REFERENCES

Sargent, R.G. (2015). An Introductory Tutorial On Verification And Validation Of Simulation Models. In *Proceedings of the 2015 Winter Simulation Conference*. Pp.1729–1740

Sargent, R.G. (2007). Verification and Validation of Simulation Models. In *Proceedings of the 2007 Winter Simulation Conference*. Pp. 124–137.

Balchi, O. (2004). Quality Assessment, Verification, And Validation Of Modeling And Simulation Applications. In *Proceedings of the 2004 Winter Simulation Conference*. Pp. 246–249.

Jain, A., Robinson, D., Dilkina, B., Fujimoto, R. (2016). An Approach to Integrate Inter-Dependent Simulations Using HLA With Applications To Sustainable Urban Development. In *Proceedings of the 2016 Winter Simulation Conference*. Pp. 1218–1229.

Benjamin, P., Patki, M., Mayer, R. (2006). Using Ontologies for Simulation Modeling. In Proceedings *of the 2006 Winter Simulation* Conference. Pp. 1161–1167.

Robinson, S. (1997). Simulation Model Verification and Validation: Increasing the Users' Confidence. In *Proceedings of the 1997 Winter Simulation Conference*. Pp. 53–59.

Zamyatina, E., Mikov, A., Lanin, V. (2018) Automation of Simulation Steps Using Ontological Approach. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (IC3K 2018). Pp. 223–230.

Zamyatina, E., Maltcev, G., Mikov, A. (2017) Simulation of Routing Algorithm for SDN and SON Networks. In: *2017 IEEE 11th International Conference on Application of Information and Communication Technologies*. Vol. 2. IEEE. Pp. 79–84.

Zhiyong, X., Yiming, H. (2008). SBARC: A Supernode Based Peer-to-Peer File Sharing System. Department of Electrical & Computer Engineering and Computer Science University of Cincinnati. Cincinnati. Pp. 1–12.

Ngom, A., Kamara-Sangare, F. (2018). Assessing Similarity Value between Two Ontologies». In *IC3K 2018. Proceedings of the 10th International Joint Conference on Knowledge Discovery*. Pp. 343–350.

Matta, N., Ermine, J.L., Aubertin, G., Trivin J.Y. (2002). Knowledge Capitalization with a Knowledge Engineering Approach: The Mask Method. In *Knowledge Management and Organizational Memories*. Springer, Boston, MA. Pp. 17–28.