CPN Tools-based Software Solution for Reliability Analysis of Processes in Microservice Environments

Ivan V. Artamonov [1] , Alexander P. Sukhodolov [2]
Baikal State University
Irkutsk, Russia
Email [1]: ivan.v.artamonov@gmail.com, Email [2]: suhodolov@gmail.com

Abstract **- Microservices are a popular contemporary method to build scalable distributed systems (e.g. websites). It is hard to predict executable operations reliability of the future system due to the vast number of components and their nonlinear relations. In this paper we consider the operation executed by multiple microservices as a single atomic transaction. The transaction matters only if it is completed successfully. In case of an error during an execution, or due to the necessity to discard the results, the transaction must be reversed or compensated. To analyze the reliability of the execution of such an operation we introduce a dedicated software solution based on Coloured Petri-Net, CPN, Tools. Accordingly, the interactive microservices system has to be described using Coloured Petri-net terms, and the operation must comply with certain rules of business process modelling, e.g. to have an input/output point. The software solution consists of several interconnecting units which include the CPN Tools, ACCESS/CPN and PostgreSQL. These programs are controlled by an application server through the web-interface. The software solution allows the analysis to be performed simultaneously by multiple examiners due to the implication of web technologies.**

Keywords - microservice, computer simulation, reliability, CPN Tools, Coloured Petri Net

I. INTRODUCTION

Today, the service-oriented concept is considered one of the mainstream ideas of distributed applications development. At the very core of this concept, and the service-oriented programming, lays the definition of service or function as an independent software unit with an explicitly defined interface which provides a loose coupling of an application towards the external environment. For instance, a service has a property of reusability, which, coupled with stateless, allows it to be applied multiple times within various task contexts. The term “microservice architecture” can be encountered in the software engineering literature quite often during the last several years. This is a relatively new trend that represents the direct consequence of development of the insights and trends of the service-oriented concept. By “microservice” we shall mean a self-contained program which executes a set of functions and interacts with other software through a well-defined interface via common standard lightweight protocols, e.g., HTTP. The fundamental idea of microservice is independence of an environment, a loose coupling towards it, and a high level of reusability.

The reusability refers to utilization of the same microservice to solve different tasks across various contexts. A loose coupling allows to solve a task using a variety of microservices which implement the same interface.

Thus, a microservice system is well suited to a step-by-step deployment and building scalable applications, and is suitable for concurrent development. Designing a software solution in a service-oriented programming doesn’t imply building “from scratch”, being geared to an extensive use of pre-built modules. A software program built on the principle of the service-oriented programming consists of a variety of services providing each other functionality in a distributed environment. Algorithm execution in such software is implemented by the orchestration model which involves transferring the data and focus of control between microservices. Program modification is also carried out by the modification of data transmission scheme between existing microservices. The execution of a complex simultaneous operations may be performed by a large number of microservices from various domains, therein their interaction pattern isn’t bounded to a basic serial data communication like a “chain”, but instead may involve complex structures, forks, loops, hierarchical inclusions, etc. Besides, all participants concertedly execute the common task and ensure to keep task’s results. Involving of a large number of participants may undermine the overall quality of execution, primarily on its reliability, as each new operation can increase the overall risk of the system failure by bringing in the chance of inherent failure.

Thereat, the topical issue is the prediction of functional and non-functional aspects of the software being created, assuming developers know quality attributes of each microservice. One of the most important non-functional aspects is system reliability. Considering the fact that microservices are mainly used in web-programming, and the role of internet environment is growing yearly, so web-sites traffic and load also increases. Thus, the exactingness to reliability of web-software grows too.

On top of this, major reliability works on any developed product are expected to be executed during the design phase. Since the analysis of reliability may be difficult or impossible during the system runtime while the risk of and damage by the system failure can be higher compared to the single microservice failure scenario, the reliability works should be executed during the interoperability design phase.

The analysis of the current reliability estimation methods for the service-oriented software systems showed that the major part of these methods is based on formal approach, which becomes exponentially harder to use as the complexity of the evaluated system grows. Along with that, such necessary tasks as the identification of the most reliable system configuration, system fault sensitivity analysis, etc. are not being carried out at all because of estimation complexity.

The software solution described in this article is free from aforementioned weaknesses. It is based on the same conditions as the most of current methods, but is capable of providing the multifold reliability analysis results faster and easier by using the simulation of the system.

II. LITERATURE REVIEW

The reliability of a microservice system can be analyzed with two methods: as reliability of standalone service or service composition. The first case is now a trivial task, since the scope of hardware and software systems reliability analysis is thoroughly studied (e.g., in [1-3], etc.), there is ISO/IEC 25010:2011 [4] that identifies the software reliability parameters, and the “Software Reliability Engineering” aimed at studying various systems performance and related concepts is created (short profile in [5]). On the flipside, the reliability estimation of distributed and heterogeneous systems appears to be a more sophisticated work (see [6] for the list of subject-matter literature). Due to their nature, services easily form interrelating structures called “service compositions”. Service compositions, the specifics of their creation, their quality and controllability are the main objects of exposure in scientific literature on service-oriented field.

During the service compositions reliability analysis, the reliability of atomic services is suggested to be familiar to studiers. Such information can be perceived as a user experience (as in [7]) or by program testing. The methods suggested in the literature are roughly divided by two dimensions: using the mathematical or simulation modelling. In the first case, studiers are guided by the system structure analysis (path-based approach) or by calculating the reliability with the use of system state transition analysis (state-based). The review of substantive works in this field is given in [8], [9]. At their simplest, path-based methods use only the analyze based on the sequential path of data transferring between services, while complex methods propose variety of reliability calculation formulae depending on typical structures (of loops, splits, joins, etc.). The [10] and [11] have become the initial works on the services’ and business process composition structure analysis, the disclosed idea was then used multiple times, e.g., in [12], [13], [14], etc. The state transition analysis is illustrated in [15-18].

However, any of the methods described is hard to apply in real life, and its complexity often grows exponentially the more complex become the systems examined (the combinatorial explosion issue). The simulation modelling method suits well for quality assessment of the process performance and service-oriented systems, but the real life application is complicated by the lack of multi-purpose software and technologies. While the well-formalized service composition can be modelled quite precisely, the real life business processes, especially dependent on human factor, can’t [19]. Human interaction is considered in strictly formal and restricted fields like people’s availability, task performance, arrivals and so on [20]. However, a model in which a part of work is performed by a simulation tool and another part is delegated to existing business software and human-operated terminals, can give more precisely results.

Moreover, preparing the experiment for most discrete-event simulation systems can take time comparable to a distributed application prototype development, which can be also used as a simulation model. These tools also require long-term assimilation and often allow modelling of ordinary signals processed by the system. License cost of general-purpose Commercial-Off-The-Shelf tools like Arena et al. may be too high for startups or small developer groups. While open-source packages often require experience in programming languages and only few such projects pay attention in providing functional user interface [21]. Not all simulation tools are based on strict mathematical apparatus or allow interaction with third-party software as a part of a model during simulation. The proposed software package tries to solve the above problems.

III. TRANSACTION AS AN ACT OF MICROSERVICES INTERACTION

It happens a lot, that an operation executed together by services should be meet atomicity property, i.e. has to be executed completely, or not to be executed at all. Partial execution is not an option for such processes, if the system cannot execute it completely, the system must return to the initial state. The [22-24] show that such process, being executed via information systems, takes the properties of transaction atomicity from the database theory: it consists of several operations which have to be executed altogether, or not to be executed at all. In [25], it was proposed to call such process a business transaction supplied by definition: a business transaction is a coordinated state changing of two or more parties, provided each party is ready for such action and has notice of concertedly acceptation by all parties. Inasmuch as the business transaction combines the works of several heterogeneous participants and is capable of executing the processes crucial for the enterprise, the question of such cooperation’s reliability was raised in [26] and term of reliability, enclosing the features of atomicity, fault tolerance, recoverability and stability, was carried out.

What we propose – is the new method of reliability analysis of business transaction execution in microservice environment, which doesn't require special traits and is easy to learn and interpret the results. The method is realized by the dedicated software solution (hereafter - BTSS). The Petri-net machine, seriously evolved through the last 20 years, is used as the business transaction modelling language, now it possesses a variety of "extensions" including "high-level", stochastic, timed and Coloured net. The facts are supplied in [27-29], allowing to consider the Petri-nets as a suitable tool for service compositions and distributed software modelling. Preliminary mathematical justification of Coloured Petri-net machine's potential use was given by us in [30], [31], etc. and is subject for future updates. Additionally, we have defined the following terms: by the "compensating net" we shall understand an auxiliary Petri-net transition sequence, allowing the net to be returned to one of the initial states. That means that the compensating net allows to rollback the business transaction. By the "recovering net" we shall under-stand the sequence which replaces one or several recoverable transitions, moving the net to the same state it would normally reach without replacing the transitions. Formulated differently, the recovering net uses the idea of reserving the faulty function/component, which is supposed to be replaced by the call chain of other functions, where in/out places of the chain matches the in/out places of the reserved transition.

We also should identify the terms of "fault" and "failure". The "fault" de-scribes an event of a process malfunction. By the "failure" we should understand the crash followed by system's inability to function under any other conditions. Software failure, as we know it, answers the description of conventional reliability theory's term "complete failure". The business transaction failure leads to disruption of its atomicity.

IV. SOFTWARE SOLUTION DESCRIPTION

The BTSS allows to automate the following business transaction reliability calculation tasks:

- General reliability assessment upon the given information. In this case the setting of current functional and subsystem failure parameters to the model occurs. The simulation allows to determine functional and reliability aspects of whole system.
- Alternative variations of the system realization reliability comparison. The analysis of several parameter sets or system models is performed in order to identify the optimal structure of future system.
- Sensitivity analysis. Some operations assumed to affect the business transaction reliability more than the others. The sensitivity analysis allows to identify the degree of said operations influence, and to compare various operations' cross impact.

The BTSS addresses the issue of general reliability assessment of any estimated business transaction of any complexity, given the information on its microservices reliability (performance capability, functionality) is known. With such information at hand, the reliability of the system can be calculated analytically, however, while the system complexity grows, the deployment of any formal methods becomes more complicated and less accurate. The BTSS solves the problem by the use of simulation modelling.

The methodological basis of the BTSS is the business transaction reliability estimation algorithm which includes the following operations:

1) *Business transaction structure analysis*. Identification of the microservices and their functions.

2) *Defining targeted qualitative and quantitative reliability metrics*. Identification of the guideline values of said metrics, if needed. Defining exanimated operations and their reliability levels.

3) *Constructing the business transaction model using coloured Petri-net* in CPN Tools environment (colour sets, variables and functions description, net scheme development (places and transitions), defining the initial marking).

4) *Initiating the new experiment* in BTSS: loading the file to the simulation environment, defining the experiment name, internal executions count, estimated transition firings count before the business transaction exit, reliability analysis type.

5) *Selecting the net* being analyzed (if there are more than one) and business transaction's final place. Selecting the processes being examined, setting their reliability parameters and methods of reliability improvement by recovering or compensating nets.

6) *Executing* the business transaction simulation. Simulation can consist of several execution series with different parameters to test system's sensitivity. Each series includes repeated process runs via Monte Carlo method.

7) *Simulation modelling results analysis* by: reliability metrics with given system parameters, system reliability's sensitivity towards examined operations' reliability.

If the reliability analysis goals aren't attained in a single iteration, the process structure is being altered and steps 3 to 7 are being repeated. It is assumed that testing the Petri-net model for adequacy and consistency is performed by the examiner early.

The BTSS consists of several interrelated parts (Fig. 1):

- The scheme designer is realized by the CPN Tools package. The package is distributed under the terms of GNU GPL v2. The scheme designer is a user interface for CPN Tools. Detailed package description is provided in [32] and on the official project website (see http://cpntools.org/start).
- The user interface is written on HTML which provides the ability of high-level formatting of interface elements, using CSS for styling and a number of JavaScript programs for user interaction and realization of asynchronous interaction with the application server. Thus, the interface part is intended to be run in the environment

capable of working with these languages, e.g., in a web browser.

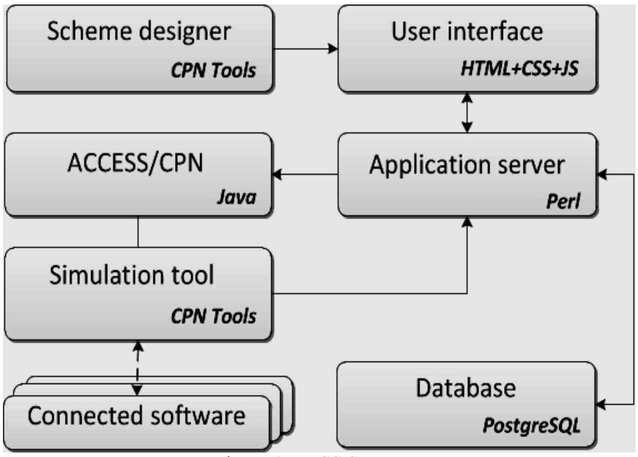

Figure 1. BTSS Structure

The application server is written on Perl. It serves the user requests, analyzes simulation data and performs statistical processing, interacts with database server. The database server is represented by the open-source DBMS PostgreSQL. The server provides storage and processing of users' data and their experiments.

- The simulation control tool is the set of two separately executed programs. The first one is written on Perl, performs library and application server function calls, and is used to execute the second program and to control the simulation. The second one is written on Java and performs the simulator initialization and execution using the ACCESS/CPN library.
- ACCESS/CPN is a part of CPN Tools, a library written on Java by the developer of CPN Tools to control it from third party software systems [33].
- The simulation tool is incorporated in CPN Tools package and is developed using the Standard ML language. It includes the SML compiler and allows to schematically declare the subroutines written on this language and to execute them during the simulation. This feature offers the possibility to realize the sophisticated behavior logic of the BT participants, e.g., to call third party programs by establishing the network connections.

Apart from CPN Tools, the BTSS also uses the jQuery (MIT license) and Highcharts (Creative Commons Attribution-NonCommercial 3.0 License) frameworks.

The BTSS is the distributed application system, capable of functioning under different operating systems and on various hardware. To ensure that, the BTSS is divided into 4 independent parts: scheme designer, user interface, database server, application server with simulator and external programs. Each part can interact with others via network, e.g., the Internet. The designed capability to expand the ability to control the simulation during concurrent operations on multiple simulators running on multiple computer systems, is implemented to ensure the scalability.

The interface, together with application server and database server, represents a three-tier client-server application capable of operating without the access to CPN Tools, e.g., to analyze the previously run simulations.

All of the mentioned above provides the ability to realize the BTSS in a form of a website where the scheme design and the simulation process control is performed by clients via the website interface, and the data processing, simulation execution and interaction with third party programs is performed by server side software. The website can be allocated in the network environment of an enterprise, institution, as well as on any of internet servers. The application and database server's ability to handle multiple clients simultaneously allows using the BTSS for academic activity or to serve several system designers at once.

The result of the business transaction simulation is represented by an analysis report, which provides the examiner with the ability to assess the business transaction reliability level.

Let us show examples of the analysis results. For instance, the total reliability analysis result is displayed in a statistical tabular form (Fig. 2). Reliability measures (atomicity, probability of faultless work, mean time between failures etc. Their definitions were given by us in [26]) show the result of simulation series and have four values: min, max, average and moda.

	Mode	Min	Max	Avg
Trasaction atomicity	0.9000	0.7600	1.0000	0.9005
Faultlessness probability	0.9000	0.7600	1.0000	0.8989
Failure frequency	0.1000	0.0000	0.2400	0.1011
MTBF	9.0000	3.1667	49.0000	11.2707
Failure rate	0.1111	0.0204	0.3158	0.1153

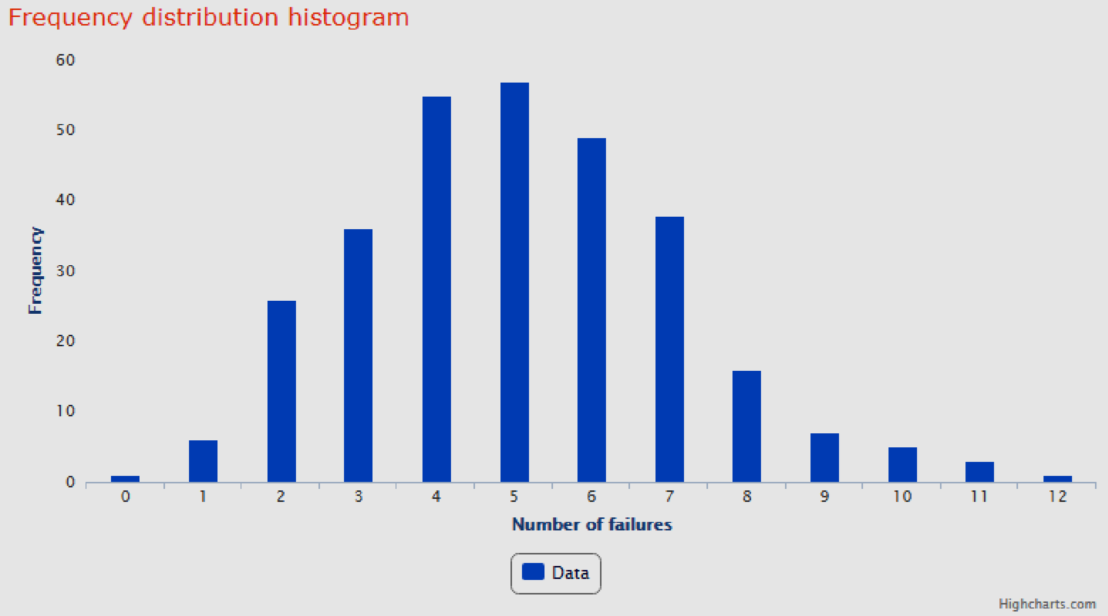

Figure 2. An example of the total result of a business transaction analysis

The separate operation analysis result is displayed in a form of generated text and statistical information table (Fig. 3). The table on Fig. 3 shows reliability measures for a particular operation not a whole process.

Process "OPERATOR" summary

Reliability indicators

	Mode	Min	Max	Avg
Fault avoidance probability	1.0000	1.0000	1.0000	1.0000
Faultness probability	0.9000	0.7600	1.0000	0.9013
Failure frequency	0.1000	0.0000	0.2400	0.0987
MTBF	9.0000	3.1667	49.0000	12.0675
Failure rate	0.1111	0.0204	0.3158	0.1126

Figure 3. An example of separate process reliability indicators table

The business transaction sensitivity towards separate operation failure analysis result is displayed in a form of a correlation diagram, where the functional connection between operation failure and business transaction reliability characteristics is being identified. Fig. 4 shows two examples of a correlation diagram. The first part demonstrates linear dependence between reliability of a whole process and some its operation, the latter shows complex, non-linear dependence.

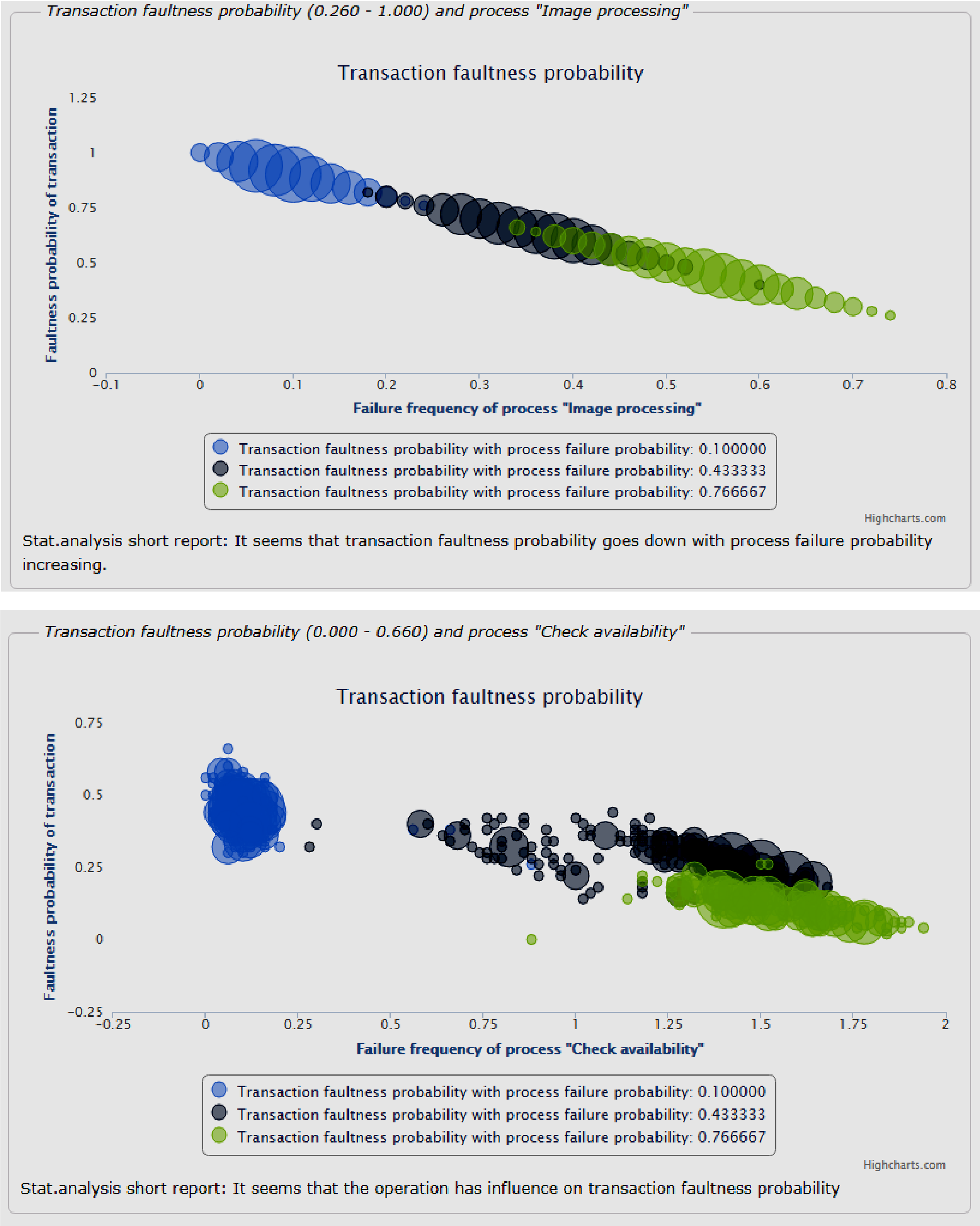

Figure 4. Two examples of atomicity sensitivity with 3 levels of process failure.

V. RESULTS AND DISCUSSION

The main advantages of BTSS over universal common means of simulation modelling are:

- Fixation on object domain. The BTSS is suited for studying the reliability of business processes of any complexity, regardless of their properties. The term "business transaction" only makes sense for the factor of "atomicity". If the examiner doesn't set a challenge of studying the atomicity, the examined process may not have the metrics of transaction.
- Easy to learn. The examiner doesn't have to scrutinize the modelling language, additional libraries or methods of signal generation. The graphic editor based on Coloured Petri-net is easy to learn and rather user-friendly.
- Ability to use the complex datasets. The majority of simulation tools makes it possible to manipulate basic signals with numerical values, only few allow to model of several basic signals at the same experiment. The BTSS uses all the advantages of Coloured Petri-net, capable to use complex datasets to perform various operations.
- Unrestricted model complexity. The CPN Tools simulator used in BTSS makes it possible to schematically identify the models of third party application calls written on any programming language to realize the complex data processing logic. In a same manner the human beings can also be engaged in the study to evaluate the human factor and get around restrictions described in [19]. Thence the business process model complexity evaluated by BTSS is unrestricted.

The sole BTSS downside is the result of its advantages: the examiner has to master the CPN Tools graphic modelling piece and to know peculiarities of Coloured Petri-net. Mastering it by microservice system developers, however, can be easier than studying and using general-purpose simulation tools or the complex formal methods of reliability analysis of service-oriented structures.

VI. CONCLUSIONS

The paper has introduced the major ideas lied on the basis of BTSS and shown the examples of analysis result. Proposed software and analysis technics are still in beta testing. The main area of future development touches Timed Coloured Petri nets that gives a possibility to test performance [34] of technological [35] or business processes [36], business transactions and micro-service systems.

REFERENCES

[1] D. Peled, D. Gries, and F. Schneider, Software Reliability Methods, Springer, 2001.
[2] P. Hoang, System Software Reliability, Springer, 2006.
[3] J. Musa, Software Reliability Engineered Testing, New York: McGraw-Hill, 1998.
[4] ISO, "ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models", 2011.
[5] M. R. Lyu. "Software Reliability Engineering: A Roadmap," FoSE 2007: Future of Software Engineering, 2007, pp. 153-170, doi:10.1109/FOSE.2007.24.
[6] W. Ahmed and W. W. Yong, "A survey on reliability in distributed systems," Journal of Computer and System Sciences, vol. 79, Dec. 2013, pp. 1243–1255, doi:10.1016/j.jcss.2013.02.006.
[7] Z. Zheng and M. R. Lyu, "Personalized Reliability Prediction of Web Services," ACM Trans. Softw. Eng. Methodol., vol. 22, 2013, pp. 12:1-12:25, doi: 10.1145/2430545.2430548 2013.
[8] N. Singh and K. Tyagi, "A Literature Review of the Reliability of Composite Web Service in Service-Oriented Architecture," ACM SIGSOFT Software Engineering Notes, vol. 40, Feb. 2015, pp. 1-8, doi: 10.1145/2693208.2693237.
[9] A. Immonen and D. Pakkala, "A survey of methods and approaches for reliable dynamic service compositions," Service Oriented Computing and Applications, vol. 8, iss. 2, Jun. 2014, pp. 129-158, doi: 10.1007/s11761-013-0153-3.
[10] J. Cardoso, J. Miller, A. Sheth, and J. Arnold, Modeling Quality of Service for Workflows and Web Service Processes, LSDIS Lab, Computer Science, University of Georgia, 2002.
[11] W. T. Tsai, D. Zhang, Y. Chen, H. Huang, R. Paul and N. Liao, "A software reliability model for web services," in Proceedings of the Eighth IASTED International Conference on Software Engineering and Applications, M. H. Hamza, ed., Cambridge, MA, United States, Nov. 2004, pp. 144-149.
[12] J. Ma and H. Chen, "A Reliability Evaluation Framework on Composite Web Service," 2008 IEEE International Symposium on Service-Oriented System Engineering, pp. 123-128.
[13] J. M. Ko, C. O. Kim, and I.-H. Kwon, "Quality-of-service oriented web service composition algorithm and planning architecture," Journal of Systems and Software, vol. 81, iss. 11, Nov. 2008, pp. 2079-2090, doi: 10.1016/j.jss.2008.04.044.
[14] C. Xie and J. Ren, "A Dynamical Reliability Prediction Algorithm for Composite Service," Mathematical Problems in Engineering, vol. 2014, p. 10, doi: 10.1155/2014/917903.
[15] A. Strunk, "An Algorithm to Predict the QoS-Reliability of Service Compositions," 2010 6th World Congress on Services, Miami, FL, 2010, pp. 205-212, doi: 10.1109/SERVICES.2010.49.
[16] K. Peng and C. Huang, "Reliability Evaluation of Service-Oriented Architecture Systems Considering Fault-Tolerance Designs," J. Applied Mathematics, 2014, p. 1-11, doi: 10.1155/2014/160608.
[17] H. Elfawal-Mansour, A. Mansour, and T. Dillon, "Composite web QoS with workflow conditional pathways using bounded sets," Service Oriented Computing and Applications, Jun. 2013, vol. 7, iss. 2, pp. 101-116, doi: 10.1007/s11761-012-0109-z.
[18] H. Yi, C. Jiang, H. Hu, K. Cai, and A. P. Mathur, "Using Markov-Chains to Model Reliability and QoS for Deployed Service-Based Systems," 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops, Munich, 2011, pp. 356-361, doi: 10.1109/COMPSACW.2011.66.
[19] W. van der Aalst, "Business Process Simulation Survival Guide" in Handbook on Business Process Management, J. vom Brocke and M. Rosemann, Eds., 2015, pp. 337–370, doi: 10.1007/978-3-642-45100-3_15.
[20] A. Greasley and C. Owen, "Modelling people's behaviour using discrete-event simulation: a review", International Journal of Operations & Production Management, Mar. 2018, vol. 38, No. 5, pp. 1228-1244 doi: 10.1108/IJOPM-10-2016-0604
[21] G. Dagkakis and C. Heavey, "A review of open source discrete event simulation software for operations research", Journal of Simulation, Aug 2016, vol. 10, No. 3, pp. 193-206, doi: 10.1057/jos.2015.9
[22] M. P. Papazoglou, "Web Services and Business Transactions," World Wide Web: Internet and Web Information Systems, vol. 6, Mar. 2003, pp. 49–91, doi: 10.1023/A:1022308532661.
[23] M. Little, "Transactions and Web Services," Communications of the ACM, Oct. 2003, vol. 46, No. 10, pp. 49-54, doi: 10.1145/944217.944237.

[24] B. Haugen and T. Fletcher, Multi-Party Electronic Business Transactions.
[25] I. V. Artamonov, “Business Transactions: Characteristics and Distinctive Features,” Business Informatics, No.2 (20), 2012, pp. 29-34.
[26] I. V. Artamonov, “Business Transactions Reliability in Service-Oriented Environment,” Conf. Information Innovative Technologies, Prague, 2013, pp.12-19.
[27] P. Massuthe, Operating Guidelines for Services, University Press Facilities, 2009, p. 266.
[28] K. Wolf, “Does My Service Have Partners?,” Transactions on Petri Nets and Other Models of Concurrency II, Kurt Jensen, Wil M.P. van der Aalst, Eds., Springer, 2009, pp. 152–171, doi: 10.1007/978-3-642-00899-3_9.
[29] B. Zafar, Conceptual Modelling of Adaptive Web Services based on High-level Petri Nets, De Montfort University, 2008, p. 188.
[30] I. V. Artamonov, “Using the Coloured Petri-Net for Business Transaction Modelling in Service-Oriented Environment”, Irkutsk State Economic Academy (ISEA) Bulletin, 2013, No.5, p. 25.
[31] I. V. Artamonov, “Using the Coloured Petri-Net for Service Composition Modelling,” NSUEM Bulletin, No.2, 2013, pp.180-187.
[32] K. Jensen and L. M. Kristensen, Coloured Petri Nets: Modelling and Validation of Concurrent Systems, Springer, 2009, p. 384.
[33] M. Westergaard. “Software I Maintain,” Internet: https://westergaard.eu/2012/06/software-i-maintain/, Jun. 2012 [May 06, 2013].
[34] L. Sokolova, “Genesis of "productivity" category”, Bulletin of Baikal State University, 2003, vol.3-4, pp. 52-55.
[35] A. Borovsky and D. Sachkov, “Methods and algorithms of CAD development for designing automated control systems of technological processes”, Modern technologies. System analysis. Modeling, 2016, pp. 119-126.
[36] T. Kubassova and N. Kleschenko, “The base strategies of development the internet-shops in modern conditions”, Logistics in the conditions of economic turbulence, 2017, pp. 65-68.