Emulation Software for Enterprise Network Data Collection at the Stage of DSS Development for EOR Implementation at Oilfields

Iakov S. Korovin* and Maxim G. Tkachenko
Southern Federal University, Taganrog, Russia

*Corresponding author

Abstract. Frequently, at the stage of decision support systems adaptation, oriented to real-time data analysis, coming from distributed systems of an enterprise computer network, connection between the DSS and working equipment is unavailable at early stages of software development. Such connection is usually available at final development stages after successful alpha testing on simulation models, confirming the declared operating parameters of software and hardware DSS complexes. For development, research and modeling of new approaches to data analysis, it is necessary to perform multiple starts of the system in conditions, close to the real ones. In the paper, we represent a method, which provides access to historical data of industrial equipment operating, using intermediate program layer. The layer simulates data input and allows the user to define the data load rate, equal to the same rate of real processes or several times exceeding it.

1. Introduction
Simulation modelling of an all-in-one complex, which collects and pre-processes data, received from the computer network of the enterprise level automatic control systems, is a complicated problem, goes beyond the scope of DSS development for analysis of real-time drilling and field exploitation data. However, owing to the cloud DSS architecture, developed by the authors [1-6], and the existing copy of telemetric monitoring historical measures database, available at early stages of development, we can set bounds to the simulation model of the enterprise hardware complex, according to the model of a data storage system (DStS), which can be filled with the real-time data, received from real industrial objects. Such organization of the DStS allows simulation of the enterprise data collection and storage, and does not require implementation of the all-in-one functioning model of all objects of the complex. Development of the DStS simulation model, which simulates sequential data loading, we describe further in the paper.

2. Development of the Simulation Model of A Data Storage System
To perform the model functions, which simulate functioning of the computer complex, we have decided to be limited to modelling of the data storage subsystem of the industrial computer network. The general structure, which shows interaction of the developed DSS modules and components with existing enterprise firmware complexes via the enterprise DSS, is given in Fig. 1. According to the structure the main interaction of the DSS components and the firmware facilities of the enterprise is based on interaction with the centralized database of the enterprise DStS.
In this solution direct connection to the other control system (CS) of the enterprise is not supposed because of complexity and variety of such systems, used in every industrial complex of the enterprise level. At the same time, the developing DSS is oriented to a wide range of solving tasks and assumes capability of efficient integration into the structure of a wide range of industrial enterprises. Use of the approach with limited access to the centralized database will further allow adaptation of the developed DSS for other enterprises of oil-gas producing industry and other ones.

To create the DSS simulation model, on the stage of development and laboratory testing of the suggested methods, we have decided to replace direct access to the enterprise DSiS by a simulation program module, which contains functions, such as creation of a working database (DB) for modelling needs, and sequential filling of the working DB with historical data, stored in a copy of the DSiS tables. The working DB is filled at the rate of the data adding into the DSiS DB according to the time marks of making measures, stored in the DB [7]. Embedding of an intermediate unit will provide safe functioning of industrial equipment during the DSS development process, owing to access separation and work with a copy of industrial measurements database. Besides, this solution will allow simulation modelling on data, given by the customer at early stages of software development, and will require no modification of access methods to the DSS data.

So, to realize simulation models, it is necessary to transform the structure of the DSS and the enterprise CS interaction (see Fig. 1). Besides, the structure is to be completed with a simulation model of the enterprise DSiS, and a stored copy of the DSiS historical data. The modified structure is given in Fig. 2.

The proposed model can be effectively used for experimental research at the early (alpha- & beta-) stages of software development, but its implementation requires the formation of a manual or automatic copy of the database of historical measurements. This solution is hard to implement in final stages of software integration into industrial operation process.
Since in most cases of industrial implementation the optimal implementation of the DSS model requires maintaining the possibility of operational integration into the structure of the industrial facility, it is extremely important to implement in the developed model support of data transfers not only with the copy of historical measurements, but also with real data stored in the DStS of the industrial enterprise.

In this case, it is not necessary to force the application to operate through the copy of the historical measurement database, and universal access interface implemented in the data gateway module can be used to access such database. Thus, the intermediate layer (database "copy of historical measurements") can be excluded from the scheme above, and the universal access interface to the database can be developed by implementing transformations of application data into data compatible with the connected database through SQL queries (Fig.3).

![Figure 2. Embedding of the simulation model of the enterprise DStS.](image)

**Figure 3.** Simulation model of the enterprise DStS with universal data access interface.

When implementing both data access schemes, all interaction between the DSS modules and the corporate database is reduced to calling the data gateway methods included in the Enterprise DStS simulation model. This approach allows to transfer the whole solution to new platforms without
changing the parameters of the SDR by modifying the data gateway connection parameters for functionally similar systems.

In case of system migration to new platforms with a different stored data structure, operator will need to re-run the automatic training and configuration procedures. In the second case, the configuration procedures can be performed using the configuration tools and graphical user interfaces provided by the DSS modules, which do not require the personnel to have in-depth knowledge of the structure and performance parameters of the individual modules.

Both data access schemes can implement the same data loading algorithms, and the use of a copy of measurements or an external enterprise DStS as a data source does not affect the operation of the proposed algorithm.

Data transfer from the existing source of the historical measures DB for further analysis of the DSS is to be performed at the rate equal to the incoming data rate in the enterprise CS, and at the considerably higher rate for speeding up modelling and evaluation of processing rate limits of incoming data.

The general algorithm of the suggested simulation model is given in Fig. 4.

Simulation starts with creation of the working tables, available to the DSS modules, which have the structure, similar to the one of the DStS data copy tables. In the case, when the working DB already contains tables with similar names, the existing tables will be renamed; an index will be added to the
name of the table to prevent loss of important data during modelling. If there is no need in saving the data, the existing tables can be automatically deleted, when modelling is started.

According to the algorithm, two time moments are fixed after creation of the tables – the beginning and the end of the simulated time interval. Re-calculation of these time marks, and sequential data copying from the enterprise DB copy into the working DB of the simulated DSS are performed repeatedly.

The suggested algorithm provides sequential data loading from the table of historical measures at a rate similar to the rate of adding data to the historical measures database. As a result, it simulates real-time functioning of the system. The algorithm contains a parameter $T_{acc}$, used for accelerating modelling processes. Its specifies the fold change for loading the historical data selection into the DSS working table. During modelling, 1 second of the modelling time corresponds to $T_{acc}$ seconds of real time of the historical data selection.

Owing to large values of the $T_{acc}$ parameter (5000-10000 and more), it is possible to simulate loading of the available historical data, accumulated during one year, in the time interval of several hours. For the majority of tasks, such performance considerably exceeds the requirements to the DSS systems. Owing to the parameter $T_{acc}$, it is possible to detect the maximum data processing rate on the computational resource of the laboratory bench or a hardware platform, granted by the customer of the DSS for testing [8].

On the base of the reviewed algorithm, an emulation software is developed. Its main form is given in Fig. 5.

![Figure 5. The task graph.](image)

The developed application allows the following settings: the parameters of connection to the data copy of the enterprise DStS, the intervals of simulation modelling, the parameter of the modelling time acceleration $T_{acc}$. Besides, it is possible to start modelling of data incoming into the DSS working tables according to the algorithm given above. After start of modelling, the data of the DB copy are sequentially copied according to the same algorithm.

### 3. Research of the Simulation Module Functioning for Historical Data Loading

In the paper, we suggest to use an additional module, which simulates an enterprise level DStS, for testing the developed DSS modules on a local computational node. Realization of the models of the enterprise CS components, which are the sources of incoming sensor data, is not required. To evaluate the time delays and DSS performance reducing, caused by use of this module, we suggest a number of test starts of the DSS on the existing copy of DStS data, which contains the historical measures of real oil Western Siberian company technological measures for the period 2017-2018 years [9-11].

To evaluate the time delays, caused by an additional module of DStS simulation, we have organized batch processing for various time intervals on a laboratory bench. Data was taken directly from the copy of the enterprise DB tables. We used a data gateway with various values of the coefficient $T_{acc}$. 


For our research, we used a laboratory bench based on the server with the Linux CentOS 7.0 operation system. The server contains 2 Intel Xeon E5-2665 processors (8 cores; the frequency - Turbo 3.1 GHz, Base 2.4 GHz), RAM - 128 Gb DDR4.

During the test starts, by means of the OS, we have registered the total operation time of the model DSS with direct access to the DStS data, with use of the suggested module of the DStS simulation modelling.

Our test series contains time measures for various values of the coefficient $T_{acc}$, specified as a parameter of the program. For the test starts, we used the coefficient values 10, 100, 1000, 200000. These values were selected according to evaluation of various acceleration of the modeling process.

The test results are given in Table 1. For the tests, we have selected the following time intervals of DStS data collection: 1 hour (test 1), 24 hours (test 2), 672 hours or 1 month (test 3) [12-13].

**Table 1. The results of simulation modelling.**

| №  | Number of records, thousand | Processing time for direct access | Increasing of the operation time of the DStS modeling |
|----|-----------------------------|----------------------------------|-------------------------------------------------------|
|    |                             | Tacc = 10                        | Tacc = 100                                           | Tacc = 1000                                           | Tacc = 200000 |
| 1  | 907                         | 5 m 30 s                         | 6 m 13 s                                             | 15 s                                                  | 13 s         |
| 2  | 20410                       | 2 hrs 10 m                       | 2 hrs 25 m                                           | 19 m 31 s                                             | 8 m 45 s     | 5 m 20 s |
| 3  | 551310                      | 58 hrs 40 m                      | 68 hrs 20 m                                          | 6 hrs 34 m                                            | 1 hrs 12 m   | 24 m 30 s |

The test results prove that owing to small values of the coefficient $T_{acc}$, it is possible to test the developed DSS in conditions close to the real ones. If it is necessary to make analysis on the base of the data, collected during a large time interval, it is possible to use large values of the coefficient $T_{acc}$.

In such a case, the time delays, carried in by the DStS simulation module, will considerably depend on data transfer costs, artificial time delays that simulate data incoming, and will not influence the process of simulation modelling [14].

According to the test results, the time costs, caused by the simulation module for data transfer, do not exceed 4.1% from the operation time of the DSS computational modules in comparison with the DSS direct access to the DB on the laboratory bench. Such increasing of the processing time is not critical for test starts.

Despite the fact that experimental studies show that the implemented algorithm introduces insignificant time delays, for the industrial operation of the system an important factor is the minimization of unproductive delays. In order to reduce the negative impact of the algorithm it was proposed to develop a database access interface in the form of two closely related versions of the software product - research and industrial.

In this case, the research version of a product should contain the subroutines of realization of the artificial delays at data transmission demanded for carrying out of experimental researches on imitation models, including test applications in industrial field. In the industrial version of the module at the level of the source code of the subprogram of making delays should be excluded, for example, with the use of preprocessor directives. In this case, module configuration parameters related to time delays ($T_{acc}$) will be ignored and data conversion overheads will be excluded.

The general scheme of data conversion with the use of the proposed algorithm assumes the one-time execution of a number of procedures of setting up and launching with the subsequent multiple launch of commands from the side of the software, contributing or extracting data from the connected storage. This situation requires the use of methods of building an application with maximum optimization of data conversion areas, including the proposed algorithm procedure. Refusal from such a solution for debugging purposes or for other reasons in most cases will not allow to achieve the results obtained in the test runs described above.

Within our project, we performed simulation modelling of the suggested module with various values of the acceleration coefficient $T_{acc}$. The obtained results prove that adaptation of the data module increases the total operation time of the system by 4.1% for large values of the acceleration coefficient (200000). For smaller values, the operation time of the application additionally increases.
by a value, equal to the time interval of historical data collection diminished by factor of $T_{acc}$. It fully corresponds to the task, which is modelling data incoming to the enterprise DStS.

4. Conclusion
In the paper, we have represented an approach to adaptation of the special-purpose software for modelling of an enterprise-level data collection network for solving tasks of DSS components simulation modelling at the development stage. Owing to the suggested approach, it is possible to use the data of historical monitoring, obtained during a large time interval, for evaluation of the DSS occurred during drilling and operation of oil wells.

Acknowledgments
The paper is published due to financial support of the Ministry of Science and Higher Education of Russian Federation, project unique identifier RFMEFI57517X0152.

References
[1] I. Korovin, A. Kalyaev. Development and research of the ANN based model for increasing oil production. ACM International Conference Proceeding Series, March 2019, Pages 107-110.
[2] Hassamien, A.E.a, Elfattah, M.A.b, Aboulenin, S.b, Schaefer, G.c, Zhu, S. Y. d, Korovin, I.e Historic handwritten manuscript binarisation using whale optimisation. SMC 2016 - Conference Proceedings. February 2017, Pages 3842-3846.
[3] Melnik, E.V., Klimenko, A.B., Schaefer, G., Korovin, I.S. A novel approach to fault tolerant information and control system design. ICIEV 2016, November 2016, Pages 1161-1164. 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016; Nabab Nawab Ali Chowdhury Senate Bhaban (Senate Auditorium), University of Dhaka; Bangladesh.
[4] Korovin, I., Melnik, E., Klimenko, A. A recovery method for the robotic decentralized control system with performance redundancy (Conference Paper). Lecture Notes in Computer Science (including subsseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 9812 LNCS, 2016, Pages 9-17. 1st International Conference on Interactive Collaborative Robotics, ICR 2016; Budapest; Hungary; 24 August 2016 to 26 August 2016; Code 179979.
[5] Korovin, I.S., Tkachenko, M.G., Khisamutdinov, M.V., Kalyaev, A.I Artificial intelligence hybrid methods application in the task of the heavy oilfields profitability increase (Article). Neftyaneo Khoozyaystvo - Oil Industry. Issue 1, January 2016, Pages 106-109.
[6] Korovin, I.S., Khisamutdinov, M.V. Neuronetwork decision support system for oilfield equipment condition online monitoring (Conference Paper). Advanced Materials Research, Volume 902, 2014, Pages 409-415.
[7] David Mascharka, Philip Tran, Ryan Soklaski, Arjun Majumdar. Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning. CVPR 2018 pre-print.
[8] Korovin Y.S., Tkachenko M.G., Kononov, S.V. Operational diagnostics of oil-field equipment based on intelligent data handling technology. Oil Industry-2012, -vol. 9, -pp. 116-118.
[9] Rabunal Ju.R., Dorado Ju., Pazos A., Rivero D., Rules and generalization capacity extraction from ANN with GP, Lecture Notes in Computer Science, Springer-Verlag Gmb, 2003. Vol. 2686, pp 606-613.
[10] Mishchenko V.A., Korobkin A.A. Using genetic algorithms for training neural networks. Modern problems of science and education2011, -Vol 6, pp. 116-119.
[11] Tenenev V.A. The solution of multicriterion optimization with genetic algorithms. Intelligent systems in manufacturing-2006, -Vol. 2, pp. 103-109.
[12] I.S. Korovin, M.V. Khisamutdinov. The Application of Evolutionary Algorithms in the Artificial Neural Network Training Process for the Oilfield Equipment Malfunctions’ Forecasting. Atlantis Press, Part of series Advances in Intelligent Systems Research, 2nd International Symposium on Computer, Communication, Control and Automation, ISBN 978-90786-77-91-8, ISSN1951-6851, DOI https://doi.org/10.2991/3ca-13.2013.63, pp 253-257.
[13] Rutkowski L. Methods and techniques of artificial intelligence. - Moscow: Hotline-Telecom, 2010. — P. 520. —ISBN 5-9912-0105-6.

[14] UCI Machine Learning Repository, URL:http://archive.ics.uci.edu/ml/