Reengineering the "Oil and Gas of Russia" Software to Meet the Requirements of Current Resilience Studies in the Gas Industry

G. Danilov[*]

Melentiev Energy Systems Institute of Siberian Branch of Russian Academy of Sciences, Irkutsk, Russia

Abstract — **Research into energy models often has to rely on the functionality provided by legacy software that cannot correctly satisfy the needs of the researcher due to the request to perform tasks that were not provided by the developer during the creation of these products. To solve these problems, reengineering of legacy software is used. In particular, it was decided to reengineer software to solve the problem of flow distribution in the gas transmission system (GTS) and find bottlenecks in it with the aid of geographic information technology. The main disadvantages of the legacy software were the inability to add a new calculation scheme or modernize it, and the inability to use geocode. The reengineering methods applied involve: 1) analysing legacy software for module connectivity; 2) choosing a framework with the capability to develop geographic information systems; 3) reworking the architecture of the software module to implement the function of adding objects of the calculation model; 4) integrating software with an embedded database management system for compactness and the ability to save changes made. These solutions made it possible to update the software and give the researcher a convenient tool for creating and correcting calculation schemes using geographic information technologies for help. Geocoding support enables the expansion of the existing energy models by adding objects based on the data provided by geographic information services for the requested tag and the research to determine new optimal solutions for transporting energy resources.**

Index Terms: **geographic information system, energy system, resilience, vulnerability.**

---

[*] Corresponding author.
E-mail: dangleb@isem.irk.ru

I. Introduction

The resilience of an energy system (ES) is its property to withstand large disturbances by preventing their cascade development with a massive disruption of energy supply to consumers and to restore the initial state of the system or bring the system close to its initial state [1–3].

The modern scheme for studying the ES resilience [4], (Fig. 1) is based on the investigation and analysis of systemic capabilities for adaptation to large disturbances and recovery after their impact [5, 6], quantitative assessment of the ES resilience [7], and the development of strategies for its increase [8–14].

As seen in Figure 1, analysis of vulnerability [15] and risks [16] plays a central role in the study of ES resilience. Resilience (Figure 1) characterizes the size and scale of negative consequences for the entire system because of the impact of a particular disturbance [17].

The initial data for the analysis of vulnerability are: information on the functioning and expansion of ES, classes of disturbances, and measures to improve resilience.

Information on the ES functioning and expansion, which includes natural-climatic and socio-economic data from monitoring energy facilities, is the result of traditional general energy research [18].

Disturbances are divided into the following classes:
- Natural disasters such as floods, earthquakes, hurricanes, etc. [19];
- Technogenic disasters caused by failures of components or subsystems [20];
- Deliberate (intentional) violations such as terrorist acts, cyber-attacks, etc. [21];

The main purpose of the ES vulnerability analysis is to identify deficiencies in the design and control mechanisms of the system, which can contribute to the spread of a large disturbance over the systems under study [17].

Vulnerability analysis of ES [24] begins with building a set of possible states of energy systems, which reflect the most representative or characteristic combinations of external conditions of their expansion and functioning in the time interval at issue. Also, disturbance scenarios describing their impact on the ES are formed according to the specified classes. Further, vulnerability analysis is divided into types, such as an analysis of global
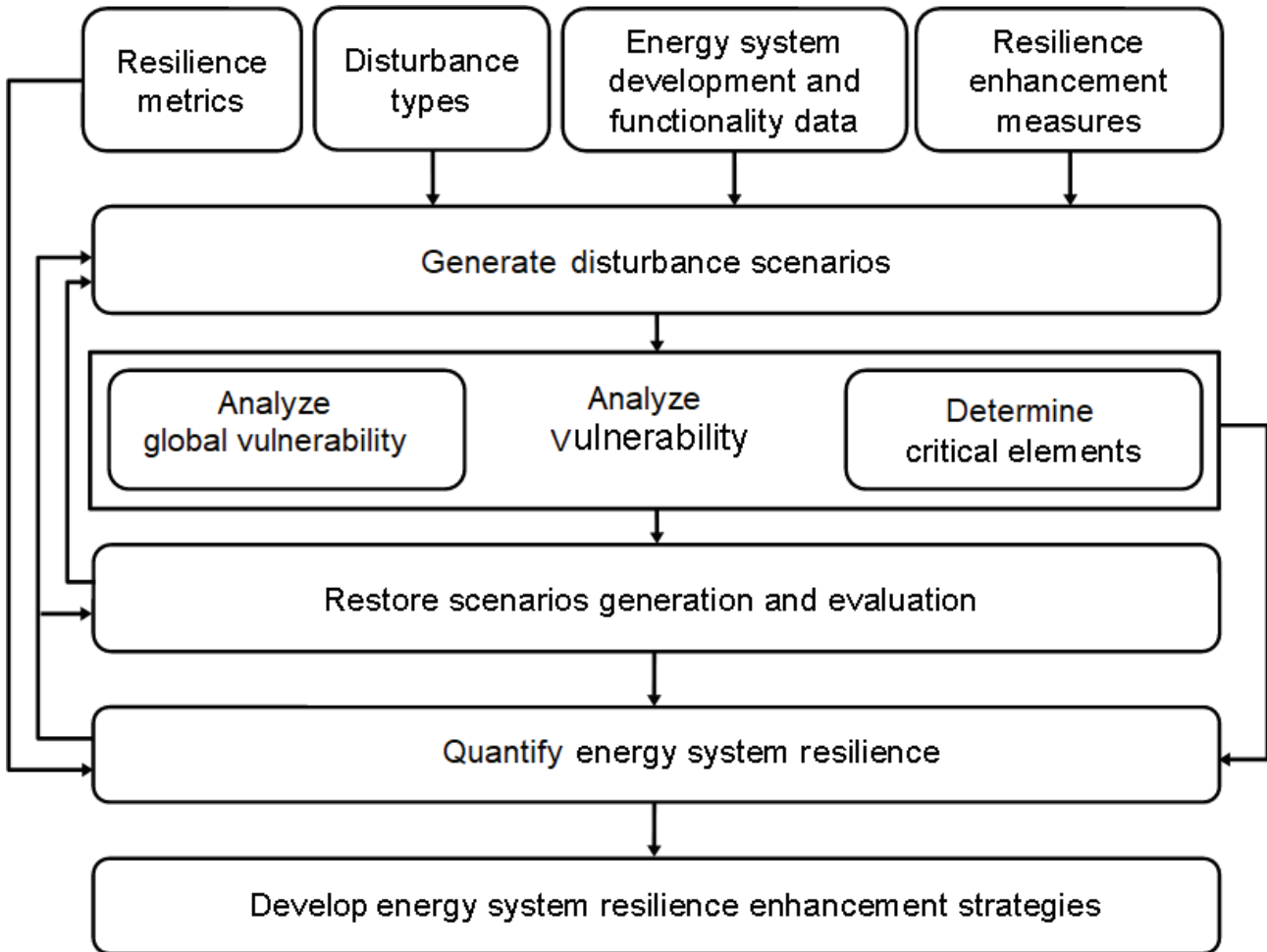
Fig. 1. General scheme of the ES resilience research.

vulnerability and a search for critical components [15, 20].

Global vulnerability analysis aims to obtain general information about the impact of disturbances on the ES performance and is carried out by modeling a series of disturbances with a gradually increasing degree of impact and, accordingly, an increasing number of consequences for the system. Such computational experiments will make it possible to determine the threshold values of the impact for certain classes of disturbances, the excess of which will cause the considered ES to split into unrelated parts [15, 20].

The search for critical components focuses on identifying the components, whose single or group failure causes the greatest decrease in the efficiency of the entire ES. The key point here is to detect all, even unexpected, sets of critical components [20].

To analyze the survivability of a unified gas system in emergency situations, the L. A. Melentiev Energy Systems Institute of the Siberian Branch of the Russian Academy of Sciences has developed the "Oil and Gas of Russia" software, which includes models that simulate the behavior of the unified gas system (UGS) and oil products systems (OPS).

II. Software "Oil and Gas of Russia"

The software is based on a flow distribution model designed to assess the production capabilities of the UGS under various types of disturbances. These studies aim to minimize possible gas shortages for consumers, identify sections of the network with insufficient capacity, which occurs under given conditions.

To solve these problems, the distribution flow in the model follows the Basaker-Gowan algorithm as a minimum cost maximum flow problem, where energy system can be represented as a graph with constraints on the throughput in arcs, and on the production, storage, and consumption volumes at the network nodes. The architecture of this software package is shown in Fig. 2.

The software, consisting of a user interface, a database, and business logic, builds a graphical image of the network, including the arcs, which represent network sections; the nodes, which stand for production, storage, and consumption facilities; and branch points where compressor stations are located in an initial ES. In addition, it makes it possible to visually display information about
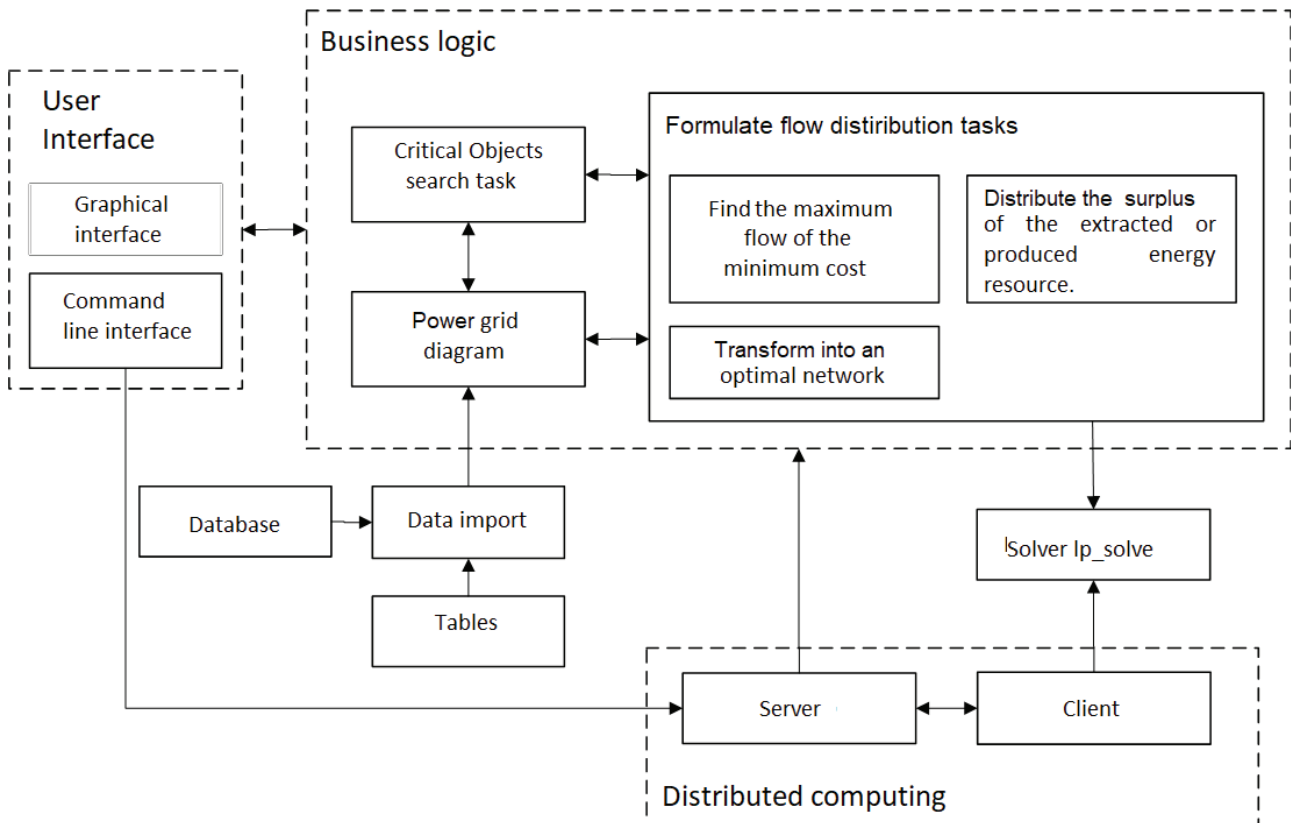
Fig. 2. Components of the "Oil and Gas of Russia" software.

each facility on the screen (volumes of production, storage, consumption at the respective nodes, and throughput in any network section belonging to the corresponding pipeline).

Before processing the database, the user needs to run the calculation program. The calculation program generates a file of results. Next, a settings file is created for the database. At the same time, the data is checked for completeness and compliance of the database files with each other and with the results file, and diagnosed errors are displayed in parallel on the screen. Only after such an analysis, the user can (or cannot) create graphic images on the screen. Diagnosed errors do not always negate the possibility of displaying graphic images. Thanks to this, it is possible to deal with each error and bring such a complex system as the UGS or OPS to the desired state (or develop another aggregation option). Several windows can be opened on one screen, i.e., for example, "Oil" and "Gas," a diagram before and after the flow distribution calculation can be shown simultaneously. The graph corresponding to the performed flow distribution differs from the original one in that the arcs and nodes involved in the calculation are shown in a different color on it. For example, the red color of the arc means that the limiting capabilities of the site have been used, the blue color means that there is no flow along the arc at all, the red color of the node indicates an undersupply of the energy resource to this node [23].

III. Analysis of the Software Package for Reengineering

The work with the software involves entering information into the database by using existing database management systems (DBMS), and importing data from excel tables or exporting model data from the database. Then, by running a separate script, the software can create a transitional configuration csv file; send the file to the calculation and computing module, using the data analysis interface; and obtain the information necessary for the study. If there is a need to change, add, or delete information, the researcher will have to re-enter the DBMS and repeat the whole cycle again. Legacy libraries are used to compile the program interface, to work with strings and display a visual interface, these methods are not supported by new versions of programming languages and do not allow switching to new development standards. Thus, reengineering encounters the following problems:

- The user wastes time due to the inability to change the system during the analysis from the interface;
- Direct interaction with the DBMS increases the risk of adding non-standardized information;
- Due to unnecessary information in the DBMS, software cannot create a transitional configuration file which the calculation and computing module depends on and without which will not be able to work correctly;
- The lack of support for methods implemented by legacy libraries makes it difficult to use software on new operating systems.
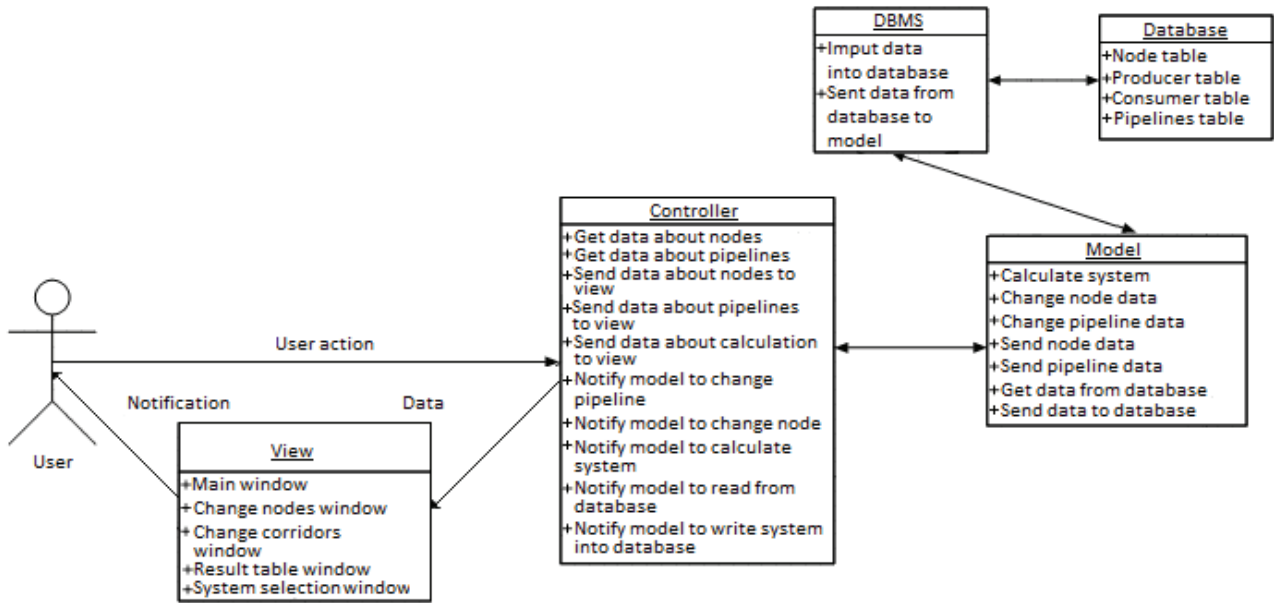
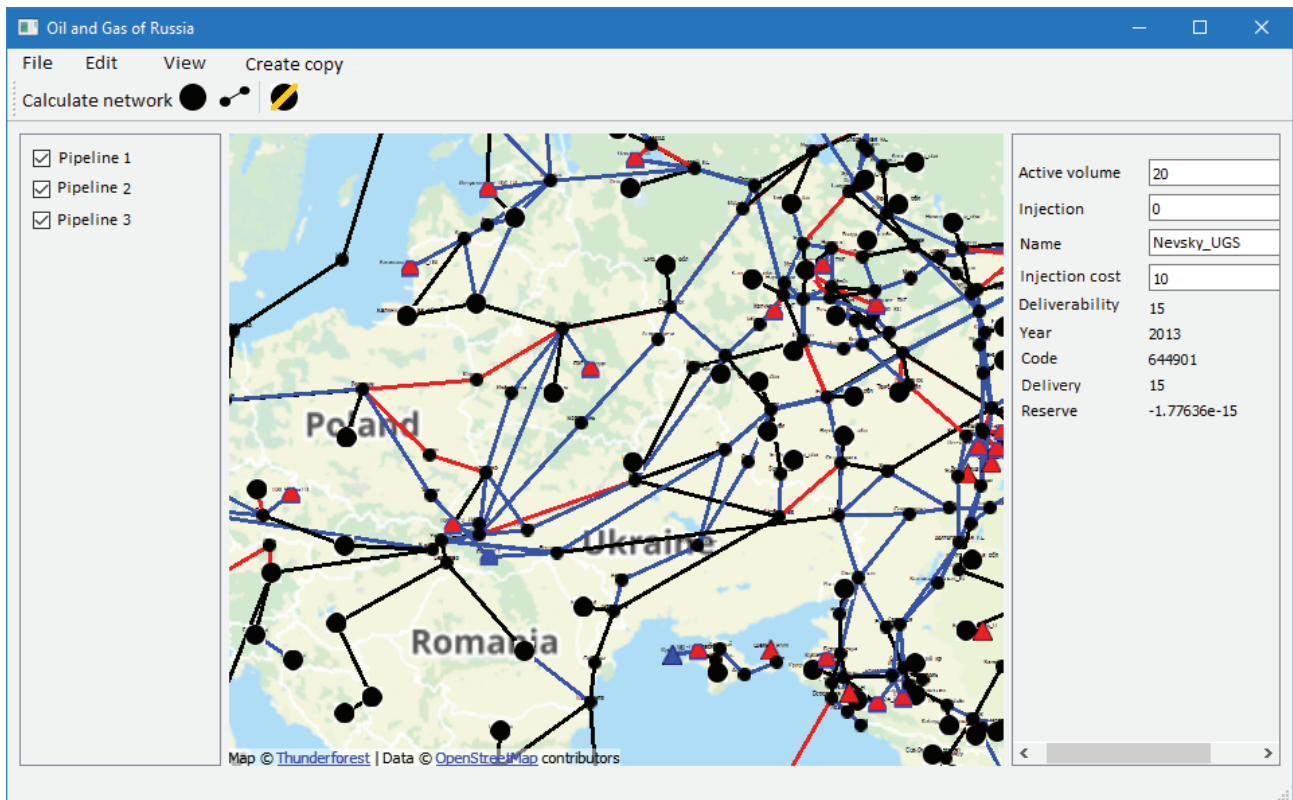Fig. 3. New architecture of the "Oil and Gas of Russia" software.

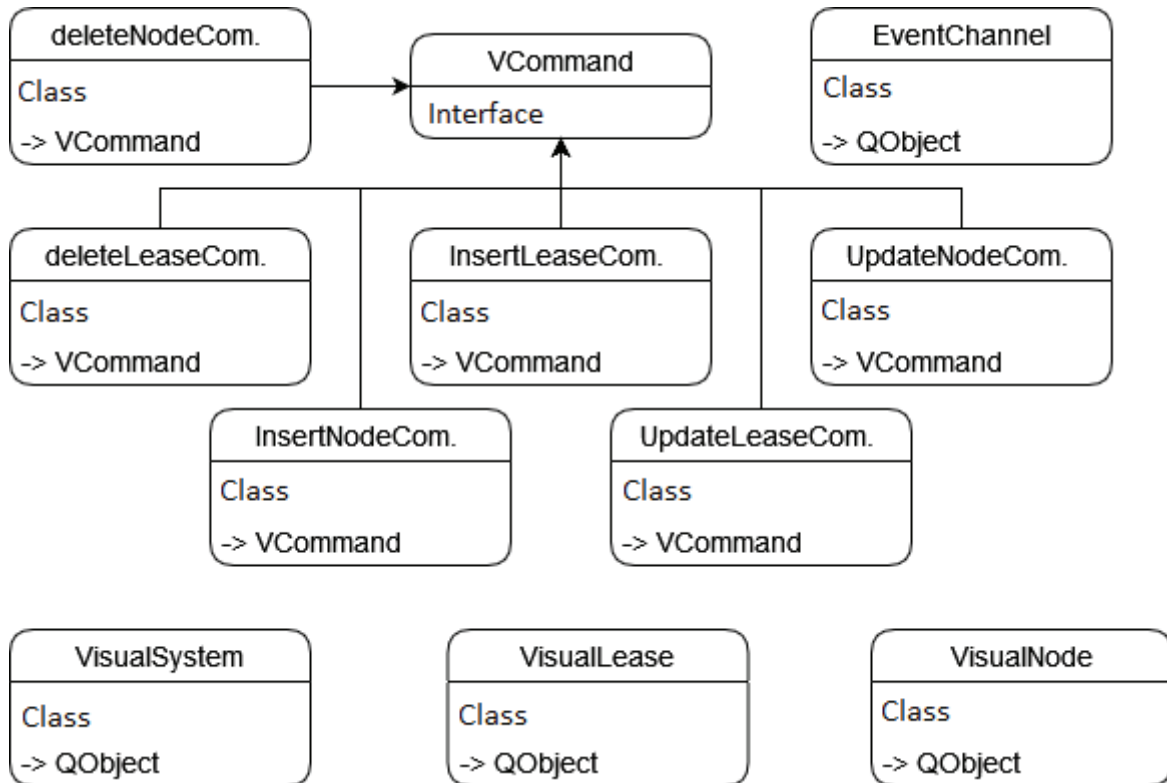Fig. 4. View of the prototype dedicated geographic information system.

Fig. 5. A scheme of classes corresponding to the topological model of the metasystem.

IV. Software Package Reengineering

One of the architectural patterns for reengineering was the Model-View-Controller pattern, which divides data, user interface, and control logic into three separate components: model, view, and controller.

This architecture makes it possible to separate the business logic component associated with the creation of an energy system from user interface and visually represent the model (graph) on map by encapsulating it in as a graph model that sends information needed for representing ES on map, thereby removing the problem of business logic dependence on inherited components. The graph contains only information about the ES facilities necessary for construction, including the type of the facility, geographic data, its name and main characteristics. The controller is used to communicate with the business logic (model). It allows customizing the data flow and standardizing information for display in a topological model. Each model uses its controller, which sets up its data flow. This makes it possible to use different types of models without changing the graph representation for each model. An option of the new architecture is shown in Fig. 3.

To switch to this architecture and reengineer the application, the Qt framework was chosen. This environment supports the C++ programming language, which is necessary to simplify the refactoring of an application for a new architecture, and also has several distinctive features that simplify the development of this architecture:

- Meta-object system. This feature of the framework allows using signals and slots for software development. In this case, when developing a view, one can use the mechanism of slots and signals to develop behaviour, receive and send information from the application, and then use the controller to determine and interpret this information for business logic, which facilitates the development of MVC and MVVM applications [25].
- Libraries for working with geoinformation data and displaying it by using geocoding. This library allows using the meta-object system to display objects on the map and to perform actions with them, for example, find out the distance between two nodes.
- Libraries for working with the SQLite database.

The Model-View-Controller architecture allows components to be modified independently of one another. This will subsequently simplify refactoring one of the components under the new software development standards (Fig. 5).

To work with business logic, the EventChannel controller class is used, which receives requests from the environment to receive information from the model or enter it into the model, thereby not binding the interface to a specific model, which allows using this class diagram (Fig. 6) to develop software applicable to other ES models in the future.

The classes deleteNodeCommand, DeleteLeaseCommand, InsertNodeCommand, InsertLeaseCommand, UpdateNodeCommand,
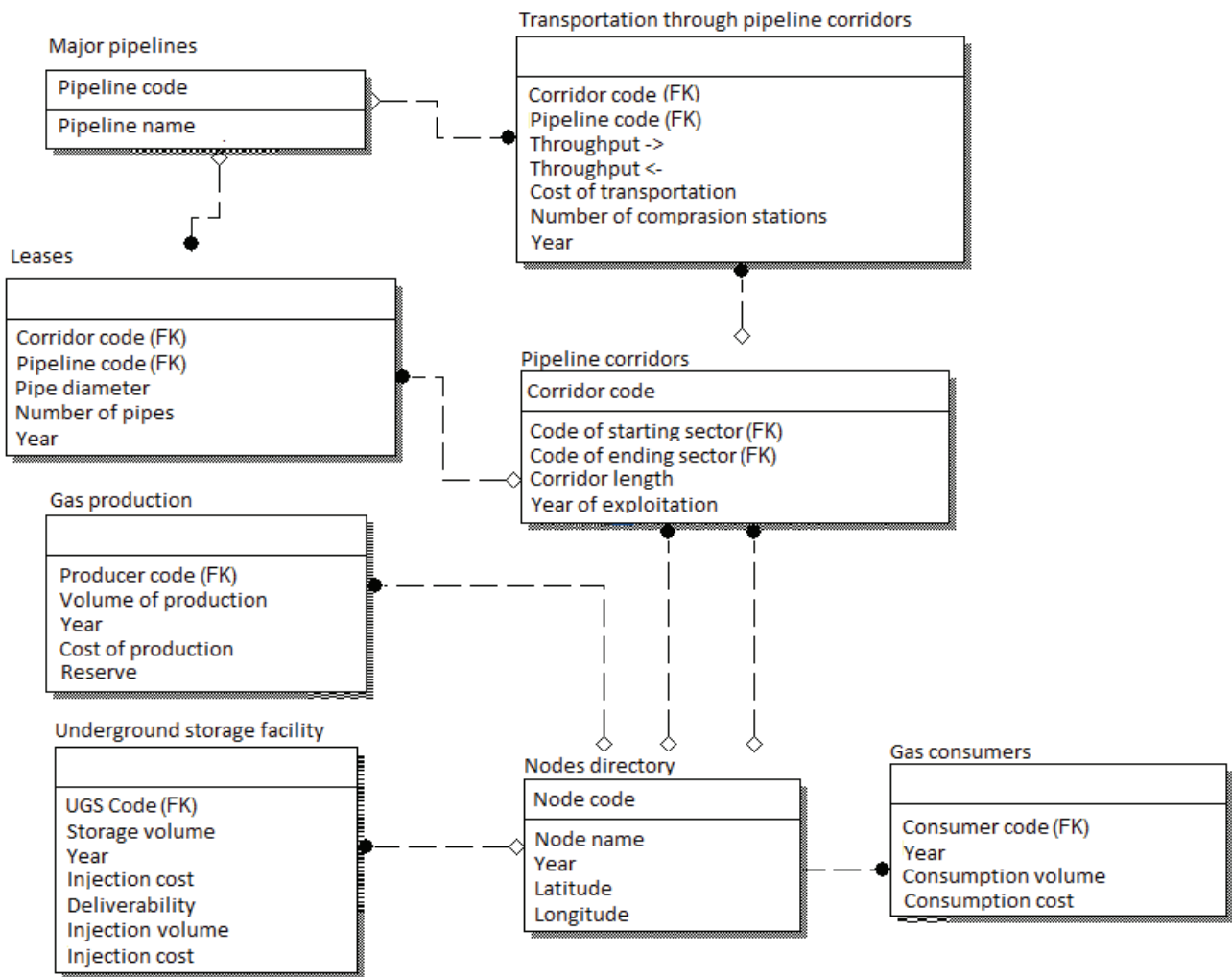
Fig. 6. Database structure.

UpdateLeaseCommand are used to create requests to add, change, or delete information from the model. These classes are developed using the "Command" programming pattern and inherit the VCommand interface, which specifies the basic functions for canceling and returning information.

The VisualSystem, VisualLease, VisualNode classes are designed to display geographic information about objects (Fig. 5), abstracting from the calculation module and storing only important information for display, position, object type.

Also, in addition to the chosen architecture, it was decided to switch from the Paradox DBMS to the embedded SQL Lite DBMS to simplify the interaction of the software with the database where the model is stored.

The proposed structure of the relational database shown in Fig. 7 reflects the explicit separation of the ES simulation into topological and functional components.

The list of database directories required to create a topological model includes:
• the directory of all nodes containing their unique codes, names, geographic coordinates;

• the directory of pipeline sections with information about the start and end nodes, a unique code, and date of commissioning.

Construction of a functional model requires the following directories:
• A directory of pipelines;
• Directories of consumers, producers and underground gas storages with their functional characteristics;
• A directory of capacity of pipeline sections.

V. Conclusion

The reengineering of the "Oil and Gas of Russia" software aims to update the legacy software of the ES resilience analysis.

The reengineering of the "Oil and Gas of Russia" software involves mainly the transition to a new architectural type Model-View-Controller. This architecture divides the program into three rare components and allows their modification independently of each other.

In addition to the transition to a new architecture, one of the modern compact relational database management systems is also employed.

Thus, the main problems of the inherited software are:
- lack of support for new versions of operating systems;
- use of legacy libraries by the program;
- lack of application portability due to client-server architecture.
- These problems have been solved and the new version provides the following features the legacy version did not have:
- capability to deploy a portable version of the application due to the embedded DBMS;
- capability to add and change the model from the program due to the transition to a new architecture;
- support by new operating systems due to refactoring of methods relying on legacy libraries.

REFERENCES

[1] N. I. Voropai, *Reliability of energy systems (collection of recommended terms)*. Moscow, Russia: Energiya, 2007, 192 p. (In Russian)

[2] N. Voropai, C. Rehtanz, "Flexibility and resiliency of electric power systems: Analysis of definitions and content," *EPJ Web Conf.*, vol. 217, 2019, Art. no. 01018.

[3] N. I. Voropay, "Prospects and problems of electric power system transformations," *Electrichestvo*, vol. 7, pp. 12–21, 2020. (In Russian)

[4] P. Gasser, P. Lustenberger, M. Cinelli, W. Kim, M. Spada, P. Burgherr, S. Hirschberg, B. Stojadinovic, and T. Y. Sun, "A review on Resilience Assessment of Energy Systems," *Sustainable and Resilient Infrastructure*, vol. 6, no. 5, pp. 273–299, 2019.

[5] G. Huang, J. Wang, C. Chen, C. Guo, and B. Zhu, "System resilience enhancement: Smart grid and beyond," *Frontiers of Engineering Management*, vol. 4, no. 3, p. 271, 2017.

[6] S. Espinoza, M. Panteli, P. Mancarella, and H. Rudnick, "Multi-phase assessment and adaptation of Power Systems Resilience to natural hazards," *Electric Power Systems Research*, vol. 136, pp. 352–361, 2016.

[7] P. E. Roege, Z. A. Collier, J. Mancillas, J. A. McDonagh, and I. Linkov, "Metrics for energy resilience," *Energy Policy*, vol. 72, pp. 249–256, 2014.

[8] Hussain, V.-H. Bui, and H.-M. Kim, "Microgrids as a resilience resource and strategies used by microgrids for enhancing resilience," Applied Energy, vol. 240, pp. 56–72, 2019.

[9] J. L. Carlson *et al.*, "Resilience: Theory and application.," *Argonne National Lab,*2012.

[10] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, "A review of definitions and measures of system resilience*," Reliability Engineering & System Safety*, vol. 145, pp. 47–61, 2016.

[11] C. Wan, Z. Yang, D. Zhang, X. Yan, and S. Fan, "Resilience in transportation systems: A systematic review and Future Directions," *Transport Reviews*, vol. 38, no. 4, pp. 479–498, 2017.

[12] Z. Bie, Y. Lin, G. Li, and F. Li, "Battling the extreme: A study on the power system resilience," *Proceedings of the IEEE*, vol. 105, no. 7, pp. 1253–1266, 2017.

[13] A. Sharifi and Y. Yamagata, "Principles and criteria for Assessing Urban Energy Resilience: A literature review," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 1654–1677, 2016.

[14] Y. Lin, Z. Bie, and A. Qiu, "A review of key strategies in realizing power system resilience," *Global Energy Interconnection*, vol. 1, no. 1, pp. 70–78, 2018. DOI: 10.14171/j.2096-5117.gei.2018.01.009.

[15] J. Johansson and H. Hassel, "An approach for modelling interdependent infrastructures in the context of vulnerability analysis," *Reliability Engineering & System Safety*, vol. 95, no. 12, pp. 1335–1344, 2010.

[16] E. Zio, "Challenges in the vulnerability and risk analysis of critical infrastructures," *Reliability Engineering & System Safety*, vol. 152, pp. 137–150, 2016.

[17] N. Voropai et al, *Hierarchical modeling of energy systems*. Novosibirsk, Russia: Academic Publishing House Geo Ltd, 2020, 313 p. (In Russian)

[18] F. H. Jufri, V. Widiputra, and J. Jung, "State-of-the-art review on Power Grid Resilience to extreme weather events: Definitions, frameworks, Quantitative Assessment Methodologies, and enhancement strategies," *Applied Energy*, vol. 239, pp. 1049–1065, 2019.

[19] H. Jönsson, J. Johansson, and H. Johansson, "Identifying critical components in technical infrastructure networks," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 222, no. 2, pp. 235–243, 2008.

[20] K. Hausken, "Defence and attack of complex interdependent systems," *Journal of the Operational Research Society*, vol. 70, no. 3, pp. 364–376, 2018.

[21] A. Edelev, A. Vorob'ev, "Methodology for identifying bottlenecks in the operation of large pipeline systems," *SOFTWARE & SYSTEMS*, no. 107, pp.174–177, 2014. (In Russian)

[22] S. Senderov, A. Edelev, S. Enikeeva, "Information support in the study of large pipeline systems," *Computational Technologies,* vol. 4, no.5, pp.30–35, 1999. (In Russian)

[23] L. Massel, D. Podkamenny, "System analysis and reengineering of legacy software," *Science and Education of Bauman MSTU,* no.4, pp. 4, 2011. (In Russian)

[24] "The future is written with Qt," *Qt documentation*. [Online]. Available: https://doc.qt.io/. Accessed on Dec. 22, 2022.

Danilov Gleb received B.S. degree in informational communication from Irkutsk State Technical University, Irkutsk, Russia in 2017. He is currently a research engineer at Melentiev Energy Systems Institute, Irkutsk, Russia. His research interests include geoinformation systems in energy research, resilience analysis of energy systems, binary-state networks in energy systems.