

Development of a software platform for the design of district heating systems

Dmitry Sokolov^{1,*} and Evgeny Barakhtenko¹

¹Melentiev Energy Systems Institute SB RAS, Irkutsk, Russia

Abstract. The paper presents a new methodological approach to developing a universal platform for the design of district heating systems. The reasons for the development of the platform are considered and a scientific statement of the problem is given. The main methodological results obtained during the development of the platform are presented. The description of the scientific and practical significance of the results is given. A new methodological approach was developed using modern information technologies. The methodological approach is based on the paradigm of Model-Driven Engineering. The essence of this paradigm is that the software is generated on the base of formal description provided by the models. The proposed approach allows one to successfully solve the problem of separation of methods for solving applied problems and models of elements of a district heating system. The paper describes the developed architecture of the software platform. A description of the architectural subsystems of the platform is presented: the computing subsystem and the graphics subsystem.

1 Introduction

For a long period, the heat economy of Russia has been developing along the path of concentration of heat loads and centralization of heat supply. An important characteristic of district heating is centralization, that is the production of heat at combined heat and power plants or at nuclear power plants, serving thousands of consumers. Heat supply has a significant impact on various factors in the life and work of people. Heat supply provides hot water supply, ventilation and air conditioning necessary to create healthy and comfortable living and working conditions for people. Many technological production processes require maintaining constant optimal temperatures and humidity in the premises of enterprises, which makes it possible to significantly increase the efficiency and improve the quality of district heating [1].

Heating assumes the presence of a heat source at the station; a heating network in the form of heat-insulated heat pipelines delivering heat energy to consumers; heat points that prepare the heat carrier of the required parameters for consumers, and consumer equipment that uses hot water in heating systems, hot water supply, ventilation, air conditioning and in technological installations.

The operating costs of a district heating system (DHS) are very different in various cases and depend on the local climate, the cost of fuel and the use of its economy, initial capital costs and the degree of centralization and automation of the heating system. However, in all cases, the operating costs of heating are significant.

The problem of optimal design of DHS is very important and its effective solution provides a significant reduction in economic costs and an acceptable quality of heat supply. There are three main cases of the problem.

1. Determining the optimal parameters for a new DHS when designing it.
2. Making recommendations for optimizing the existing DHS.
3. Determining the optimal parameters of the existing DHS as it undergoes development.

Approaches based on the decomposition of a common complex problem into a set of less complex subproblems have been developed. Subproblems that arise after decomposition [2]:

- Selection of the optimal structural configuration of the network.
- Providing the required network transmission capacity.
- Ensuring the required level of system reliability.
- Determination of source parameters.
- Calculations of the distribution of flows and heat energy.

An algorithms for solving the problem is presented in Figure 1.

2 Formulation of the scientific problem and approach to solving

Based on the performed analysis of the existing software for modelling heat supply systems, it was concluded that it is necessary to develop a single software platform. The platform should act as a base for the automated

* Corresponding author: sokolov_dv@isem.irk.ru

construction of software systems for computer modelling, calculation and optimization of DHS.

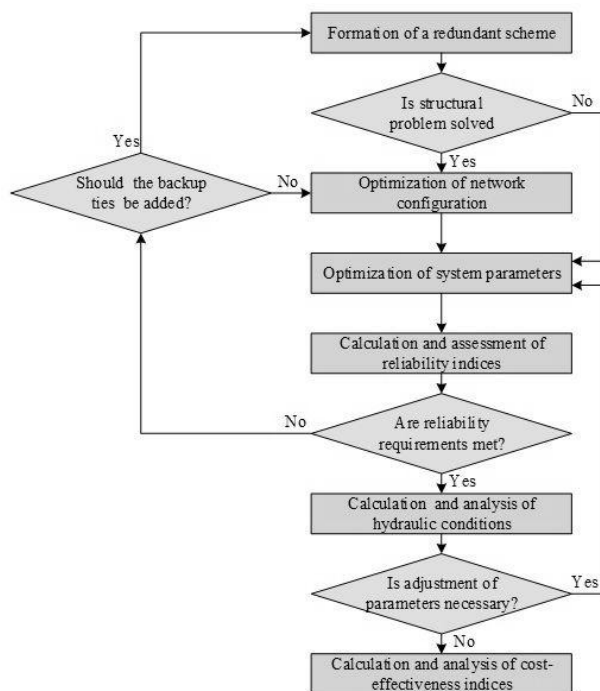


Fig. 1. A diagram of an algorithm for solving the problem.

A new scientific problem has been formulated. New methodological approach should provide the following features:

- Solving all subproblems within a single software system.
- Link the solution of particular subproblems into a single solution.
- Allows one to flexibly customize the computational procedure for the features of the equipment and mathematical models describing its operation.
- Flexibly customize the computational process by implementing various algorithms for solving the problem.

The elements of the platform should be developed in accordance with the component programming paradigm providing the ability to represent methods, algorithms and mathematical models in the form of elements of the software fund, suitable for repeated use. Software components should have software interfaces standardized within a single software platform with a description of subroutines, input and output data formats. The practical application of the component approach leads to a complex problem caused by the need to integrate into a software system a large number of software components used for modelling various elements of DHS, which can be represented by a wide range of equipment and be in different states. The high difficulty of keeping up to date the entire set of necessary software components allows us to conclude that for the implementation of the software platform, it is necessary to involve programming paradigms focused on automating the stages of software construction.

Modern approaches to software development are the research subject of many specialists, and they are widely represented in the scientific literature. Booch [3] suggests using the object-oriented approach methodology to develop complex programs. Beck [4] offers general principles for organizing effective software development. Martin [5] provides an overview of a large number of approaches to the development of general-purpose software components based on design patterns. Fowler et al. [6] propose approaches to develop complex software systems designed to solve large enterprise-level problems.

Paradigms of software design based on pre-developed models are currently of wide interest and are being actively developed by specialists. The Model-Driven Engineering (MDE) conception is an effective approach to automated software construction [7-9]. One of the directions of the development of this conception are approaches based on ontologies [10-11], which allow one to effectively describe knowledge of subject areas [12-13]. UML is proposed as a standardized language for formalizing models [14]. Approaches to the development of domain-specific languages for describing models of objects of the problems being solved are proposed [15].

MDE conception is the basis for the automated software construction based on the platform. Features of the implementation of the conception in the framework of the developed approach is using of reflective programming [16-17]. These software technologies are the development of metaprogramming [18] and provides the flexibility to dynamically construct and changes the software system.

A new methodological approach for developing a software platform for DHS design has been proposed. The main components of the approach:

- Composition of software components and their standardized program interfaces.
- Composition and methods of building ontologies.
- Three-level architecture of a multi-user software system.
- Approaches to the automated construction of subsystems of the software system.

The main components of the platform are shown in Figure 2. These are control components, components that implement mathematical methods and algorithms, components for accessing databases, ontologies (DHS ontology, problem ontology, software ontology). Java is the primary programming language for implementing the platform. Some software components may be implemented in C/C ++, Fortran, Python, Julia and other languages.

3 Architecture of the software system

The three-tier architecture of the multi-user network system is shown in Figure 3. A description of its subsystems is given below.

Tier 1. Web application running on a server. This subsystem is responsible for interacting with users within a computer network to solve computational and analytical problems on the server side.

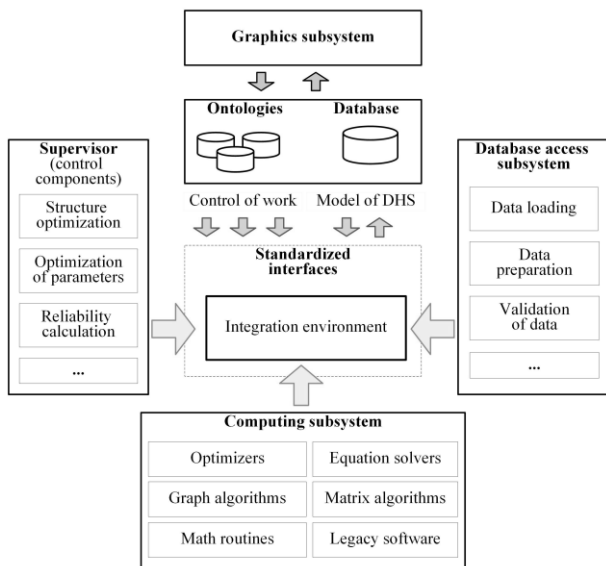


Fig. 2. The main components of the platform.

Tier 2. The client part of the network system. It is designed to work with graphic interactive models of DHS and data on their elements. This subsystem is used to prepare the initial data and view the calculation results.

Tier 3. Data storage subsystem. It is designed to store and reuse the DHS computer model, initial data and calculation results, graphic data on urban development.

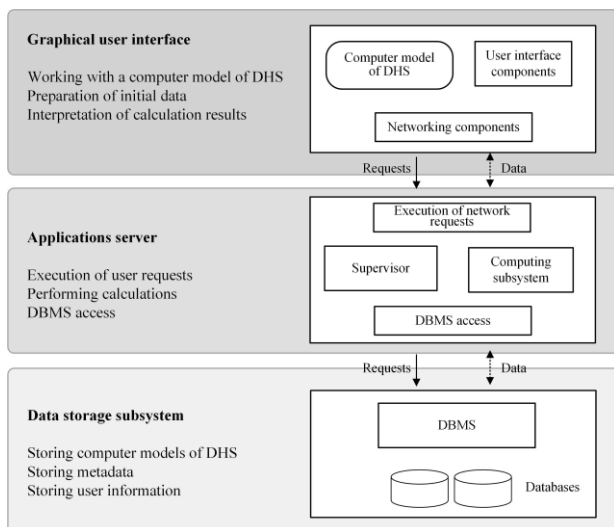


Fig. 3. Multi-user network software system. Three-tier architecture.

The server side of the software system runs under the control of the Apache Tomcat servlet container. Its architecture is shown in Figure 4. An original approach to the organization of the computing process was developed. In accordance with the approach, the construction of the computing system is carried out in an automated mode. A new algorithm for the construction has been proposed, and its steps are presented in the following form.

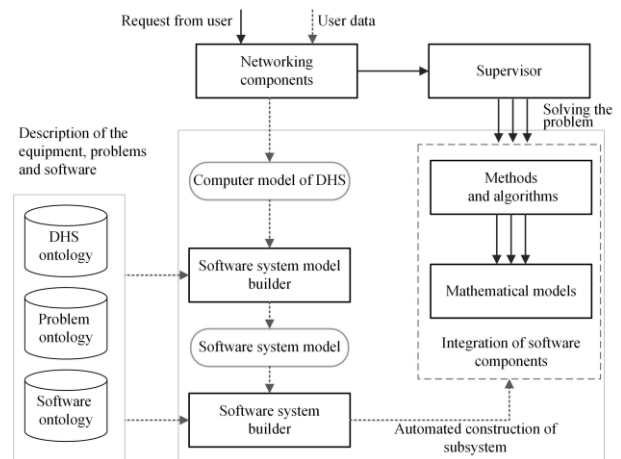


Fig. 4. Server side architecture. Automated construction on the basis of the DHS model and ontologies.

Step 1. Generate a list of used equipment using the DHS computer model created by user.

Step 2. Generate a list of mathematical models of the heat carrier flow using the DHS model.

Step 3. Generate a list of required software components using the lists created in the previous steps.

Step 4. Form a description of the software component that implements the algorithm for solving the applied problem.

Step 5. Construct the model of the software system.

Step 6. Load the description of the required software components from the software ontology.

Step 7. Connect software components to the software system that implement algorithms and mathematical models of equipment and heat carrier flow.

Step 8. Form the data structures required for the functioning of the connected software components.

The approach developed for the implementation of the client part of the system is shown in Figure 5. The software system connects the required software components in an automated mode. A new algorithm for automated construction was proposed, and its steps are presented in the following form.

Step 1. Generate a list of installed equipment using the DHS computer model created by user.

Step 2. Generate a list of graphic models using the DHS computer model.

Step 3. Generate a list of required software components using the lists created in the previous steps.

Step 4. Load the description of the required software components from the software ontology.

Step 5. Construct the model of the software system.

Step 6. Connect software components implementing graphical models of equipment to the software system.

Step 7. Form the data structures necessary for the functioning of the connected software components.

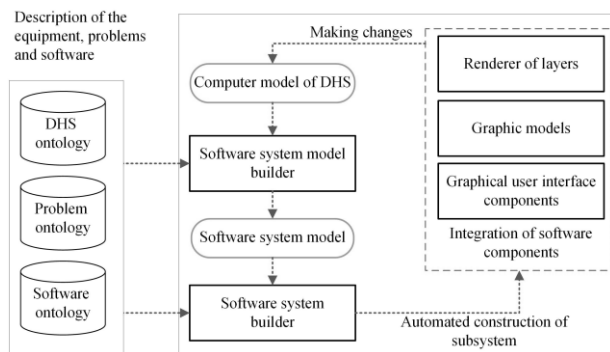


Fig. 5. Client-side architecture. Automated construction on the basis of the DHS model and ontologies.

4 Conclusions

The paper formulates a new scientific problem of developing a software platform for DHS design. An original methodological approach to the development of the platform and universal principles of its construction are proposed. The MDE conception underlies the development of the platform. Using MDE allows one to automate the process of software system construction based on the platform. The approach uses modern metaprogramming technologies. The MDE conception has been further developed within a new approach with reflective programming. The automated construction of a software system is controlled by knowledge about the subject area, the storage of which is organized in the form of ontologies. An extensible architecture of the platform is proposed. A universal approach has been developed for the integration of software components, which provides dynamic integration of mathematical models and algorithms in the context of solving an applied problem.

New efficient algorithms for solving DHS design problems are developed on the base of the platform [19-20]. These algorithms are implemented as software components and are part of the platform.

The developed methodological support can be used in organizations engaged in the development of software for the analysis, design and optimization of pipeline systems of various purposes. The implemented software, when introduced into design organizations, will increase the quality of the received design recommendations and reduce the complexity of their receipt.

The research was carried out under State Assignment, Project 17.4.1 (reg. no. AAAA-A17-117030310432-9) of the Fundamental Research of Siberian Branch of the Russian Academy of Sciences.

References

1. Z. Shiraks, *Heat supply* (Moscow, Energia, 1979).
2. V. Stennikov, E. Sennova, T. Oshchepkova, *Izv. RAN Energetika*, **3**, 44-54 (2006).
3. G. Booch, *Object-Oriented Analysis and Design with Applications* (Boston, Addison-Wesley, 2007).

4. K. Beck, *Extreme Programming Explained: Embrace Change* (Boston, Addison-Wesley, 1999).
5. R. Martin, *Agile Software Development: Principles, Patterns and Practices* (New York, Pearson Education, 2002).
6. M. Fowler, D. Rice, M. Foemmel, E. Hieatt, R. Mee, R. Stafford, *Patterns of Enterprise Application Architecture* (Boston, Addison-Wesley, 2002).
7. M. Brambilla, J. Cabot, M. Wimmer, *Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering* (San Rafael, Morgan & Claypool, 2012).
8. A. Silva, *Computer Languages, Systems & Structures*, **43**, 139-155 (2015).
9. V. Štuikys, R. Damaševičius, *Meta-Programming and Model-Driven Meta-Program Development*. London: Springer-Verlag, 2013.
10. J. Pan, S. Staab, U. Aßmann, J. Ebert, Y. Zhao, *Ontology-Driven Software Development* (Berlin, Springer-Verlag, 2013).
11. S. Staab, T. Walter, G. Gröner, F.S. Parreiras, *Reasoning Web. Semantic Technologies for Software Engineering. Lecture Notes in Computer Science*, 62-98 (2010).
12. T. Gruber, *Knowledge Acquisition*, **5**, 199-220 (1993).
13. S. Staab, R. Studer, *Handbook on Ontologies, 2nd edn* (Heidelberg, Springer-Verlag, 2009).
14. G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide, 2nd edn* (Boston, Addison-Wesley, 2005).
15. M. Fowler, R. Parsons, *Domain-Specific Languages*. Boston (Addison-Wesley, 2010).
16. B. Smith, *Procedural Reflection in Programming Languages, PhD Thesis* (Massachusetts, MIT, 1982).
17. I. Forman, N. Forman, *Java Reflection in Action* (Greenwich, Manning Publications, 2005).
18. K. Hazzard, J. Bock, *Metaprogramming in .NET* (Greenwich, Manning Publications, 2012).
19. V. Stennikov, E. Barakhtenko, D. Sokolov, *Energy* **185** (2019).
20. D. Sokolov, E. Barakhtenko, *Energy* **210** (2020).