A methodological approach to the development of a technological platform for intelligent integrated energy system modelling

Evgeny Barakhtenko[1*], and *Dmitry* Sokolov[1]

[1]Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Pipeline Energy Systems Department, Lermontov str., 130, Irkutsk, Russia, 664033

Abstract. Modeling of intelligent integrated energy systems, which has a complex structure, is performed using specialized software, which is caused by the computational complexity of the models, methods and algorithms used. The paper lists the requirements for software for modeling intelligent integrated energy systems and concludes that it is necessary to develop a technological platform. A methodological approach to the development of a technological platform for modeling integrated energy systems is presented. The methodological approach includes the following components: principles of building a technological platform, architecture of a technological platform, and the structure of an ontology system, a method of automated integration of software components into a software system.

Keywords: technological platform, intelligent integrated energy system, ontology, metaprogramming, Model-Driven Engineering.

1 Introduction

The energy infrastructure of modern cities is complex and includes fuel, electricity, heat and cold supply systems. On the one hand, these systems have some functional independence, and on the other hand, they can interact with each other in the process of functioning, using energy carriers and interchangeability of primary energy resources. This shows their natural integration, which intensifies with the development and application of intelligent, information and telecommunication technologies. The issues of creating intelligent integrated energy systems are becoming relevant [1-3].

Modeling of integrated energy systems with a complex structure is performed using specialized software due to the computational complexity of the models, methods and algorithms. The authors have developed requirements for software for modeling integrated energy systems, among which the most important are the following:

---

* Corresponding author: barakhtenko@isem.irk.ru

- modeling of integrated energy systems, which have a different composition of the types of energy supply systems and equipment, with a different set of graphical and mathematical models;
- conducting scientific research in the field of creating methodological support for solving problems of expansion planning and control of integrated energy systems;
- unified data presentation, which ensures their application in solving problems of graphical and mathematical modeling of integrated energy systems;
- performing complex engineering calculations for the optimal transformation of integrated energy systems in order to increase the efficiency and reliability of their functioning;
- solving the problems of expansion planning and control of integrated energy systems should be carried out within a single information space;
- software implementation should be based on modern information technologies and programming paradigms.

Based on the analysis of the existing software for modeling energy systems and the formulated requirements, it was concluded that it is necessary to develop a technological platform that should ensure the creation of a single information space and be the basis for the automated construction of software systems for modeling integrated energy systems. The elements of the platform should be built in accordance with the component programming paradigm, which will provide the ability to represent methods, algorithms and models in the form of elements of the software fund that are suitable for repeated use. On the other hand, the high difficulty of keeping an updated set of necessary software components requires the involvement of programming paradigms aimed at automating the stages of building software for the implementation of a technological platform.

2 Overview of approaches to software development

Existing approaches to software development are presented in [4-6]. The principles of creating universal software components are presented in [4]. The general principles of building complex software systems are considered in works [7, 8]. In [9-13], the concept of Model-Driven Engineering (MDE) is proposed for automating the stages of building software. The MDE concept is a set of methodological approaches to the automated construction of complex software systems based on previously developed models [10]. In [14-16], the UML language is proposed as a universal tool of describing subject areas.

In the process of automated construction of a software system, ontologies are used, which allow a formalized description of objects of the subject area, their properties and the relationship between these objects. Ontologies are becoming more and more widely used in solving engineering problems [17]. There are works that describe the experience of using ontologies to solve energy problems [18, 19]. General issues in the development and use of ontologies are discussed in [20, 21].

The basis for the development of methodological support and software is modern metaprogramming technologies. Metaprogramming is a type of programming associated with the creation of programs that generate other programs as a result of their work, or programs that change themselves during execution [22-25].

Reflective programming is a type of metaprogramming and is an extension of the object-oriented programming paradigm [26, 27]. Reflective programming allows one to perform operations that cannot be performed in object-oriented programming. Reflective programming allows one to track the structure of a software system and dynamically change the composition of software components during its operation.

3 Methodological approach to creating a technology platform

The authors have developed a methodological approach to creating a technological platform for modeling integrated energy systems. The methodological approach includes the following components:

- Principles of building a technological platform.
- Technology platform architecture.
- Composition and structure of the ontology system.
- Method of automated integration of software components into a software system.

The following principles of building a technological platform for modeling integrated energy systems are proposed:

- a software system that implements the preparation of a computer model of an integrated energy system, the formation of conditions for solving an applied problem and carrying out calculations, consists of graphical and computing subsystems, which are built in the context of solving an applied problem on the basis of a technological platform;
- a computer model of an integrated energy system is created graphically, and labor-intensive stages of data structure formation are automated by knowledge from ontologies;
- planning of the computational process for solving an applied problem is performed automatically on the basis of a computer model of the integrated energy system and knowledge from ontologies;
- a software system that solves an applied problem is automatically built by tools of metaprogramming in the context of solving this problem, and software components that implement mathematical models and methods (algorithms) are dynamically integrated during the calculation;
- the computing process is organized and performed automatically, while the operations of data preparation, storage and coordination of the calculation results are automated.

The architecture of a technological platform for modeling integrated energy systems includes the following main components: a graphical subsystem, a computing subsystem, a supervisor, and a database access subsystem.

Graphical subsystem is intended for working with graphical scheme of integrated energy systems and data on the elements of their schemes on the terrain plan. This system is responsible for the operation of the user with an active multi-level model of the integrated energy system on the terrain plan.

Computing subsystem is designed to solve computational and analytical problems and consists of a library of software components. The library of software components includes the following components:

- universal software components that implement methods and algorithms for solving applied problems;
- software components that implement models of elements of integrated energy systems.

Supervisor performs the following functions:

- to carry out loading of the software components necessary for modeling by metaprogramming;
- to manage the graphical subsystem;
- to control the computing process, during which the sequence of calling the required software components is organized;
- to manage the operation of the database access subsystem.

During its work, the supervisor uses knowledge, the storage of which is organized in the problem ontology and the software ontology.

Database access subsystem is designed to store and reuse a computer model of an integrated energy system, initial data and calculation results, graphical data on urban area.

E3S Web of Conferences **219**, 04003 (2020)   https://doi.org/10.1051/e3sconf/202021904003

Mathematical Models and Methods of the Analysis and Optimal Synthesis of the Developing Pipeline and Hydraulic Systems 2020

The formal description of knowledge about the subject area in the form of ontologies makes it possible to reuse them in the automation of software construction. The technological platform includes an ontological system [27], which is used to solve the following tasks: automated construction of a software system; information content of the technological platform; the use of a technological platform in solving applied problems.

The system of ontologies includes the following ontologies: metaontology, ontology of integrated energy systems, ontology of problems, and ontology of software.

Metaontology is contains basic concepts that are used in the construction of applied ontologies.

Ontology of integrated energy systems contains a description of the hierarchical structure of the integrated energy system construction, the classification of the equipment and its technical characteristics, a description of the elements and their properties, the classification and description of the mathematical models.

Ontology of problems contains a description of applied problems, methods and algorithms for their solution, parameters that are initial data, and parameters obtained as a result of solving the problem.

Ontology of software contains a description of software components and their properties, metadata (input and output parameters, a description of data formats) and a description of technologies and interfaces for accessing software components.

The method of automated integration of software components into a software system is as follows. Each element of the modeled integrated energy system is represented by a separate software component and has a unique number. The storage of these components in memory is organized using hash tables. An illustration of the work of the method of automated integration of software components into a software system is shown in Figure 1. When a user creates a new element of a computer model of an integrated energy system, the model manager allocates memory to store a data structure describing this element. Then the user enters data on the properties of this element of the integrated energy system through the appropriate tools of the graphical interface. After that, the user selects a model that describes the characteristics of the equipment and sets the state of the item. The model manager checks if the required model component is connected, which implements the user-selected model of the integrated energy system element. If the required model is not available, then it is integrated into the platform from the library of graphical components by tools of reflective programming. When a model becomes available, the model number is entered into the data structure describing the new element of the integrated energy system. Once connected to the software system, the model component can be reused to represent the elements of the integrated energy system, the type and state of which corresponds to this model.

4 Conclusion

The paper presents a methodological approach to the development of a technological platform for integrated energy system modeling. The technological platform allows one to create a single information space and is the basis for the automated construction of software systems for integrated energy system modeling. The developed methodological approach includes the following components: principles for constructing a technological platform, architecture of a technological platform, composition and structure of an ontology system, a method for automated integration of software components into a software system. Using this methodological approach, a software prototype of the platform have been implemented. The software prototype of the platform have been used in the study of real and test schemes of integrated energy systems. On the basis of the prototype, the emergency conditions on the example of the electric and heat supply systems of one of the district of Irkutsk have been investigated. The methodology of demand management of electricity and heat in an

integrated energy system with active consumers have been tested on a test scheme, for which optimal electricity and heat load profiles have been obtained. The calculations showed the possibility of using the platform with the proposed architecture for the study of integrated energy systems.
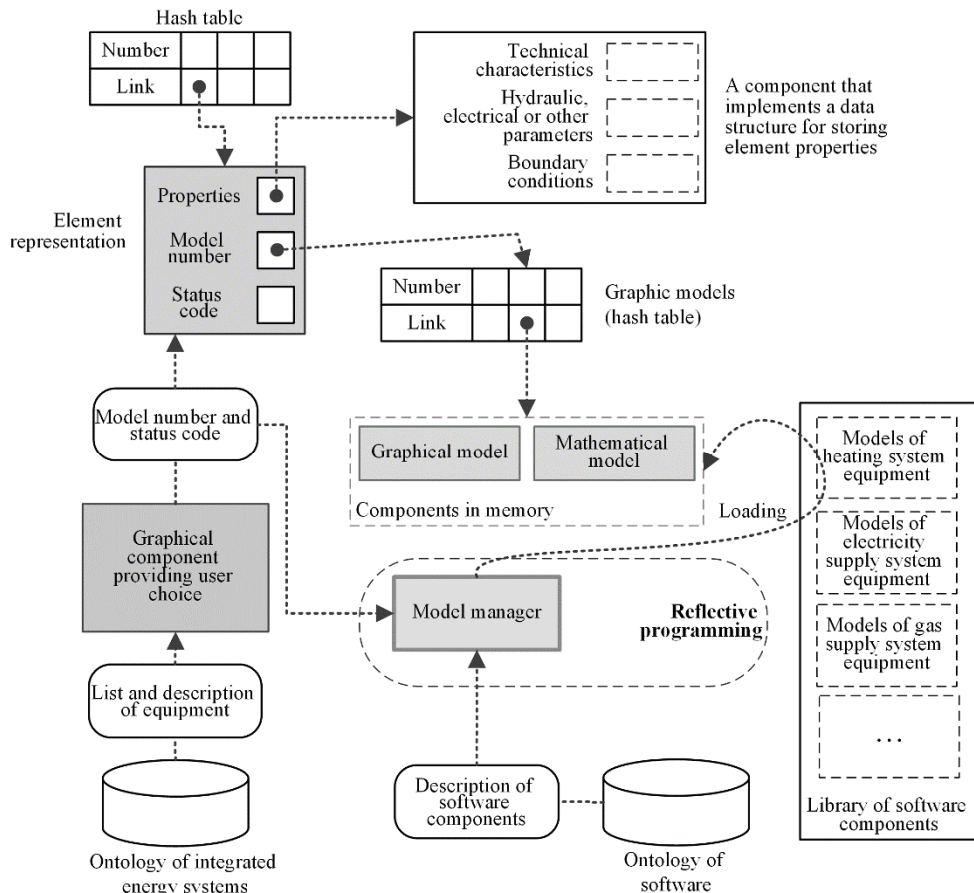

Fig. 1. Illustration of the automated integration of software components into a software system.

References

1. X. Ran, R. Zhou, Y. Yang, L. Lin, *Proc. of the IEEE Power and Energy Society General Meeting*, 1 (2012)
2. M. Geidl, IEEE Transactions on Power Systems, **22**, 145 (2007)
3. N. Voropai, V. Stennikov, S. Senderov, E. Barakhtenko, O. Voitov, A. Ustinov, Journal of Energy Engineering, **143** (2017)
4. R. C. Martin, *Agile Software Development: Principles, Patterns and Practices* (2002)
5. K. Beck, *Extreme Programming Explained: Embrace Change* (1999)
6. G. Booch, *Object-Oriented Analysis and Design with Applications* (2007)

E3S Web of Conferences **219**, 04003 (2020)   https://doi.org/10.1051/e3sconf/202021904003

Mathematical Models and Methods of the Analysis and Optimal Synthesis of the Developing Pipeline and Hydraulic Systems 2020

7. M. Fowler, D. Rice, M. Foemmel, E. Hieatt, R. Mee, R. Stafford, *Patterns of Enterprise Application Architecture* (2002)

8. M. Fowler, R. Parsons, *Domain-Specific Languages* (2010)

9. D. C. Schmidt, Computer, **39**, 25 (2006)

10. M. Volter, T. Stahl, J. Bettin, A. Haase, S. Helsen, *Model-Driven Software Development: Technology, Engineering, Management* (2006)

11. M. Brambilla, J. Cabot, M. Wimmer, *Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering* (2012)

12. A. R. Silva, Computer Languages, Systems & Structures, **43**, 139 (2015)

13. V. Štuikys, R. Damaševičius, *Meta-Programming and Model-Driven Meta-Program Development* (2013)

14. G. Booch, I. Jacobson, J. Rumbaugh, *The Unified Software Development Process* (1999)

15. G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide* (2005)

16. C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (2004)

17. G. B. Evgenev, *Intelligent Design Systems* (2009)

18. Y. A. Zagorulko, G.B. Zagorulko, Vestnik NGU, **10**, 121 (2012)

19. T. N. Vorozhtsova, S. K. Skripkin, **13**, 376-381 (2008)

20. T. R. Gruber, What is an Ontology? (2009) [Online], http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

21. N. Guarino, Understanding, Building, and Using Ontologies (2010) [Online]. http://ksi.cpsc.ucalgary.ca/KAW/KAW96/guarino/guarino.html.

22. A. P. Ershov, *Programming Programme for the BESM Computer* (1959)

23. B. C. Smith, *Procedural Reflection in Programming Languages* (1982)

24. Bartlett J. *The art of metaprogramming, Part 1: Introduction to metaprogramming* (2005) [Online] http://www.ibm.com/developerworks/library/l-metaprog1/

25. Wirth N. *Compiler Construction* (1996)

26. K. Hazzard, J. Bock, *Metaprogramming in .NET* (2012)

27. I. Forman, N. Forman, *Java computection in Action.* (2005)

28. V. Stennikov, E. Barakhtenko, D. Sokolov, Advances in Intelligent Systems Research, **158**, 152 (2018)