УДК 004.415.2

ПРИМЕНЕНИЕ МЕТАПРОГРАММИРОВАНИЯ В ИНТЕГРИРОВАННОЙ ГРАФИЧЕСКОЙ СРЕДЕ ДЛЯ МОДЕЛИРОВАНИЯ ТРУБОПРОВОДНЫХ СИСТЕМ ЭНЕРГЕТИКИ

Соколов Дмитрий Витальевич
К.т.н., с.н.с., e-mail: sokolov_dv@isem.irk.ru
Барахтенко Евгений Алексеевич
К.т.н., с.н.с., e-mail: barakhtenko@isem.irk.ru
Федеральное государственное бюджетное учреждение науки Институт систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук,
664033, Иркутская область, г. Иркутск, ул. Лермонтова, д. 130

Аннотация. В статье представлен методический подход, позволяющий средствами метапрограммирования автоматизировать построение интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики различного типа. Автоматизированное построение интегрированной графической среды выполняется в рамках концепции Model-Driven Engineering на основе компьютерной модели трубопроводной системы и онтологий.

Ключевые слова: интегрированная графическая среда, автоматизация программирования, Model-Driven Engineering, метапрограммирование, рефлексивное программирование, онтология.

Цитирование: Соколов Д.В., Барахтенко Е.А. Применение метапрограммирования в интегрированной графической среде для моделирования трубопроводных систем энергетики // Информационные и математические технологии в науке и управлении. 2019. № 1 (13). С. 96–104. DOI: 10.25729/2413-0133-2019-1-08

Введение. Эффективное решение задач проектирования и эксплуатации трубопроводных систем энергетики (ТПС) (тепло-, водо-, нефте-, газоснабжения и др.) невозможно без применения интегрированной графической среды. Эта среда представляет собой программную систему, позволяющую решать информационные (работа с компьютерной моделью ТПС и ее элементами на плане местности в интерактивном режиме), расчетные (решение инженерных задач) и аналитические (просмотр графиков, таблиц и результатов расчетов) задачи в рамках единого интерфейса пользователя.

Общность топологических свойств ТПС позволяет ставить задачу разработки единой программной системы, являющейся интегрированной графической средой для компьютерного моделирования ТПС различного типа. В проектных и эксплуатационных организациях применяется специализированное программное обеспечение, ориентированное на конкретный тип ТПС и определенные классы решаемых задач. В настоящее время отсутствует описание методических подходов к построению интегрированных графических сред, обладающих универсальностью и возможностью моделирования различных типов ТПС в рамках единой программной системы.

Высокая трудность поддержания в актуальном состоянии всего набора необходимых программных компонентов позволяет сделать вывод, что для реализации интегрированной графической среды необходимо привлечение парадигм программирования, ориентированных на автоматизацию этапов построения программного обеспечения (ПО).

Современные подходы к разработке ПО изложены в работах [4, 5, 15]. Принципы создания универсальных программных компонентов представлены в работе [15]. Общие принципы построения сложных программных систем рассматриваются в работах [10, 11]. В работах [9, 21] для автоматизации этапов построения программного обеспечения предлагается концепция Model-Driven Engineering (MDE). Концепция MDE представляет собой совокупность методических подходов к автоматизированному построению сложных программных систем на основе предварительно разработанных моделей [22]. В качестве универсального средства описания предметных областей предлагается язык UML [6, 7, 14]. В настоящее время разработаны подходы, где в качестве средства описания предметной области предлагаются онтологии [1, 2, 12, 16, 17]. Подходы к автоматизации этапов построения программного обеспечения, основанные на применении метапрограммирования, изложены в работах [9, 13, 21]. Рефлексивное программирование является одним из видов метапрограммирования и представляет собой расширение парадигмы объектно-ориентированного программирования [9, 13, 18]. Рефлексивное программирование позволяет выполнять операции, которые в рамках традиционного объектно-ориентированного программирования выполнить невозможно [9].

В статье предложен методический подход к автоматизированному построению интегрированной графической среды средствами метапрограммирования. На базе этого подхода реализована программная система, которая представляет собой программный прототип интегрированной графической среды для компьютерного моделирования ТПС различного типа.

1. Методический подход и его особенности. В результате проведенных авторами исследований предложен методический подход, в рамках которого объединены:

- концепция MDE;
- современные технологии метапрограммирования.

В рамках предлагаемого подхода MDE адаптирована к особенностям предметной области компьютерного моделирования ТПС, а автоматизированное построение программной системы выполняется на основе компьютерной модели ТПС и онтологий [3, 19, 20]. В процессе построения интегрированной графической среды используется система онтологий, состоящая из метаонтологии и трех прикладных онтологий (онтология ТПС, онтология задач и онтология ПО). Для долговременного хранения онтологий авторы отдали предпочтение формату XML, который является универсальным средством обмена данными между программными системами и предоставляет возможность организации хранения сложных (иерархических, сетевых) структур данных, а в современных платформах программирования существуют библиотеки, предназначенные для работы с этим форматом.

В настоящее время технологии метапрограммирования активно развиваются, и в рамках парадигмы объектно-ориентированного программирования развивается рефлексивное программирование, которое является основой предлагаемого методического подхода и применяется для автоматизированного построения интегрированной графической

среды. Рефлексивное программирование позволяет отслеживать структуру программной системы и динамически изменять состав программных компонентов во время ее работы.

Разработанный авторами методический подход включает:

- принципы автоматизированного построения интегрированной графической среды для компьютерного моделирования ТПС;
- архитектуру интегрированной графической среды, описывающую устройство программной системы, которая создается в автоматизированном режиме;
- методику автоматизированного построения программной системы на основе концепции MDE, рефлексивного программирования, онтологий и компьютерной модели ТПС.

Основные принципы автоматизированного построения интегрированной графической среды:

- Программная система, ориентированная на моделирование конкретной ТПС, строится в автоматизированном режиме в контексте решения задачи построения компьютерной модели этой ТПС (рис. 1).

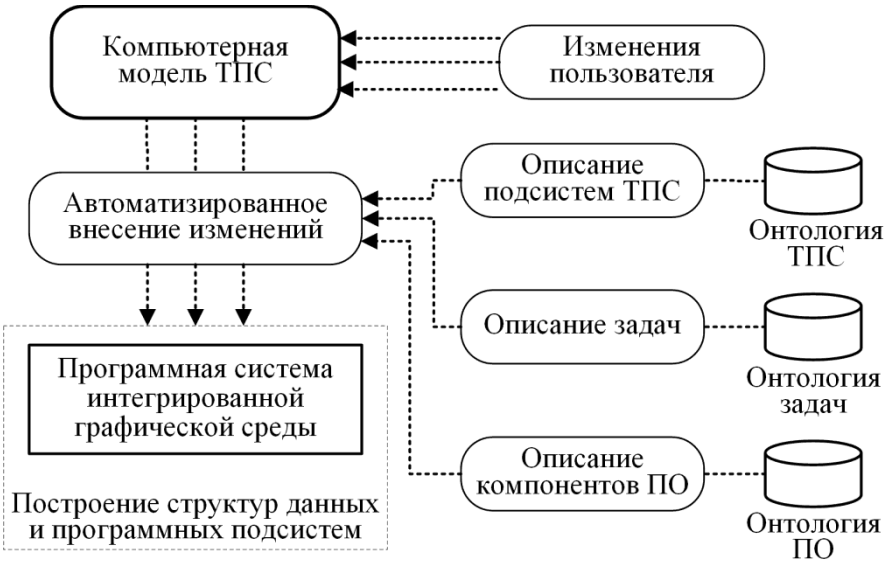

Рис. 1. Иллюстрация принципа построения интегрированной графической среды

- Автоматизированное построение программной системы и ее адаптация к конкретным особенностям моделируемой ТПС выполняются средствами рефлексивного программирования.
- Для автоматизации этапов построения программной системы и информационного наполнения пользовательского интерфейса используются знания, хранение которых организовано в виде онтологий.
- Знания, описывающие общие для всех типов ТПС свойства, задачи и используемое для их решения ПО, сохраняются в виде онтологий, а конкретные особенности моделируемой ТПС описываются ее компьютерной моделью.

Авторами предложена архитектура интегрированной графической среды (рис. 2), которая включает следующие составляющие:

- Графическая подсистема, которая обеспечивает работу пользователя с активной графической моделью ТПС на плане местности, и позволяет ему просматривать данные в удобном для восприятия виде и вносить необходимые изменения.

- Инструментальная подсистема, которая обеспечивает решение следующих задач: создает необходимый графический вид меню, позволяя пользователю решать задачи, соответствующие типу моделируемой ТПС; создает и предоставляет пользователю панели инструментов для рисования, редактирования, изменения состояний элементов активной модели ТПС; формирует диалоговые окна для настройки отображения модели ТПС, занесения необходимой информации и параметров, влияющих на решение прикладных задач; создает панели инструментов для рисования элементов плана местности и городской застройки.
- Подсистема доступа к базам данных, которая обеспечивает организацию работы с различными базами данных (БД), применяемыми для хранения и многократного использования компьютерных моделей ТПС, исходных данных и результатов расчетов, данных об объектах городской застройки.

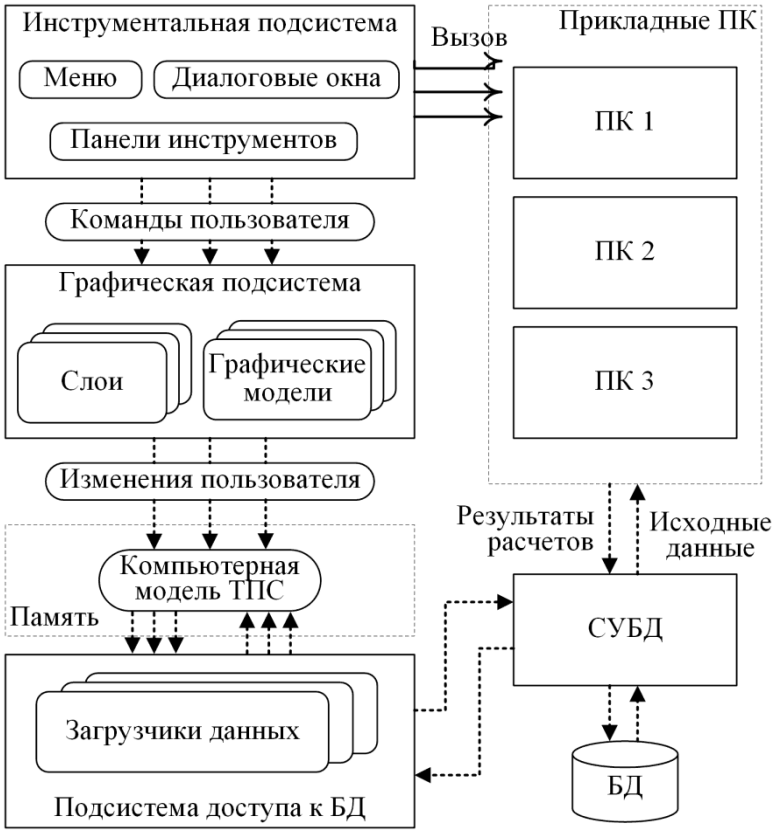

Рис. 2. Архитектура интегрированной графической среды

Представленная архитектура является описанием устройства программной системы, которая создается в автоматизированном режиме и реализует интегрированную графическую среду для компьютерного моделирования ТПС.

При реализации интегрированной графической среды в соответствии с предлагаемым методическим подходом авторы ориентируются только на свободное ПО. В качестве базового языка программирования используется Java, выбор которого обусловлен поддержкой современных технологий объектно-ориентированного, компонентного и функционального программирования, а также встроенной поддержкой рефлексивного программирования и других видов метапрограммирования [9]. Для работы с БД

используется СУБД Firebird, которая обладает возможностью многопользовательского доступа и переносимостью между различными операционными системами.

2. Методика автоматизированного построения. Согласно предложенной методике построение интегрированной графической среды в автоматизированном режиме выполняется в соответствии со следующим алгоритмом.

Шаг 1. Построение инструментальной подсистемы. На основе описания типа ТПС, соответствующего выбору пользователя, создаются элементы графического интерфейса и необходимые структуры данных.

Шаг 1.1. Формирование структур данных, описывающих состав элементов ТПС конкретного типа и их свойства.

Шаг 1.2. Создание главного меню графической среды, позволяющего пользователю решать задачи, соответствующие типу моделируемой ТПС.

Шаг 1.3. Загрузка в память графических изображений для создания элементов панелей инструментов на основе описания элементов ТПС в онтологии ТПС.

Шаг 1.4. Создание панелей инструментов, содержащих элементы графического интерфейса для работы с компьютерной моделью ТПС.

Шаг 1.5. Формирование структур данных, описывающих список возможного к установке в ТПС оборудования и набор моделей (графических и математических).

Шаг 1.6. Формирование структур данных, описывающих состав решаемых прикладных задач и используемого для их решения ПО.

Шаг 1.7. Формирование структур данных, описывающих список входных и выходных параметров для решаемых прикладных задач.

Шаг 2. Построение подсистемы доступа к базам данных. Компьютерные модели ТПС различных типов хранятся в БД, состав объектов которых соответствует составу элементов ТПС, их свойствам, набору используемого оборудования и его характеристикам. Доступ к этим моделям осуществляется через ориентированные на конкретный тип ТПС специализированные программные компоненты. Подключение этих компонентов выполняется в автоматизированном режиме, во время которого из онтологии ПО считывается описание программных компонентов, соответствующих типу моделируемой ТПС. Далее по описанию этих компонентов выполняется их интеграция в программную систему средствами рефлексивного программирования.

Шаг 3. Построение графической подсистемы. Основная идея, предлагаемая авторами, заключается в том, что графическая подсистема, отвечающая за отображение активной модели ТПС, строится динамически (в автоматизированном режиме), в ответ на действия пользователя, направленные на внесение изменений в компьютерную модель ТПС. В случае, когда пользователь создает новую модель ТПС определенного типа, интегрированная графическая среда создает базовый набор моделей элементов, соответствующих типу моделируемой ТПС. В случае, когда пользователь загружает ранее созданную модель ТПС, среда дополнительно к базовому набору моделей элементов ТПС подключает набор моделей элементов, соответствующий составу элементов моделируемой ТПС. Модель элемента (компонент-модель) представляет собой программный компонент, реализующий графическое изображение и набор алгоритмов, необходимых для обеспечения активности компьютерной модели ТПС (например, обработка действий пользователя мышью, проверка принадлежности точки области элемента, перемещение, масштабирование и др.). Однажды

подключенный к программной системе компонент-модель может быть многократно использован для представления элементов ТПС, тип и состояние которых соответствует этой модели.

3. Практическая реализация. Разработанный методический подход применен при реализации программной системы, которая представляет собой программный прототип интегрированной графической среды для компьютерного моделирования ТПС. Реализованный программный прототип ориентирован на работу с компьютерными моделями теплоснабжающих систем и позволяет создавать, вносить изменения и сохранять их в БД для многократного использования при решении прикладных задач.

Подсистемы интегрированной графической среды, отвечающие за создание инструментов графического интерфейса (инструментальной подсистемы), взаимодействие с БД (подсистемы доступа к базам данных) и работу с активной моделью ТПС (графической подсистемы), формируются в автоматизированном режиме с учетом особенностей конкретного типа моделируемой ТПС. Построение выполняется в соответствии с приведенной выше методикой.

На рис. 3 представлен вид главного окна интегрированной графической среды, которое содержит элементы интерфейса, формируемые инструментальной и графической подсистемами. Инструментальная подсистема обеспечивает работу следующих элементов главного окна интегрированной графической среды:

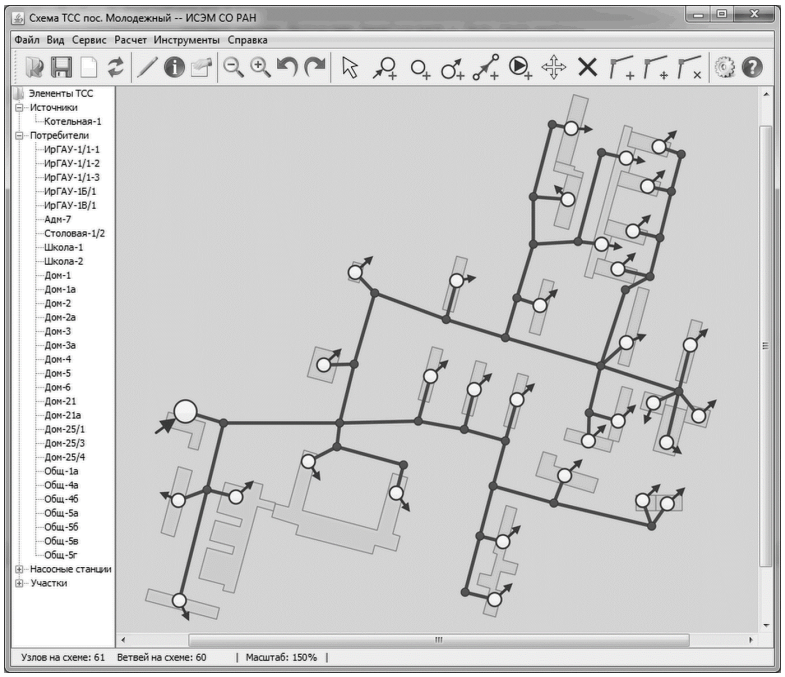

Рис. 3. Вид графического интерфейса интегрированной среды

- главное меню, которое позволяет пользователю выполнить следующие действия: работать с файлом БД; управлять отображением модели ТПС; управлять видом рабочей области окна; решать прикладные задачи моделирования ТПС;
- панели инструментов, которые обеспечивают пользователю реализацию следующих функций: работы с файлом БД; управления отображением модели ТПС; управления видом рабочей области окна; работы с элементами активной графической модели ТПС; работы с объектами городской застройки на плане местности;
- дерево схемы моделируемой ТПС для работы с ее элементами;
- строка состояния, которая отображает основные сведения о процессе моделирования ТПС.

Центральную часть главного окна занимает рабочая область, отвечающая за работу пользователя с активной графической моделью ТПС и объектами городской застройки на плане местности. За реализацию этой части пользовательского интерфейса отвечает

графическая подсистема, которая динамически адаптируется (формируется) к следующим действиям пользователя:

- загрузке в память данных, описывающих ранее созданную модель ТПС;
- созданию новой модели ТПС при выборе пользователем ее типа;
- изменениям, вносимым пользователем в компьютерную модель ТПС.

Заключение. Предложен авторский подход к автоматизированному построению интегрированной графической среды для компьютерного моделирования ТПС различного типа. Оригинальность этого подхода заключается в реализации следующих идей:

- Компьютерная модель ТПС определенного типа может быть представлена как совокупность графа, описывающего конфигурацию этой системы, и набора программных компонентов-моделей, описывающих свойства ее элементов.
- Автоматизированное построение программной системы, реализующей интегрированную графическую среду, выполняется на основе компьютерной модели ТПС и онтологий на базе концепции Model-Driven Engineering и рефлексивного программирования.

Авторами реализована программная система, которая представляет собой программный прототип интегрированной графической среды для компьютерного моделирования ТПС энергетики различного типа и назначения.

Предложенный авторами методический подход обладает универсальным характером и может быть использован в организациях, занимающихся разработкой ПО для компьютерного моделирования ТПС энергетики различного типа и назначения.

Исследования выполнены при частичной финансовой поддержке Российского фонда фундаментальных исследований (грант № 16-07-00948).

СПИСОК ЛИТЕРАТУРЫ

1. Ворожцова Т.Н., Скрипкин С.К. Использование онтологий при моделировании программного комплекса // Вычислительные технологии. 2008. т.13. ч.I. С. 376–381.
2. Ворожцова Т.Н., Скрипкин С.К. Онтологический подход к моделированию программного комплекса // Вестник ИрГТУ. 2006. №2(26). С. 72–78.
3. Стенников В.А., Барахтенко Е.А., Соколов Д.В. Разработка принципов построения интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики // Информационные технологии. 2018. №24. С. 313–320.
4. Beck K. Extreme Programming Explained: Embrace Change. Addison-Wesley. 1999.
5. Booch G. Object-Oriented Analysis and Design with Applications. Addison-Wesley. 2007.
6. Booch G., Jacobson I., Rumbaugh J. The Unified Software Development Process. Prentice Hall. 1999.
7. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide, 2nd edn. Addison-Wesley. 2005.
8. Brambilla M., Cabot J., Wimmer M. Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering. Morgan & Claypool. 2012.
9. Forman I., Forman N. Java Reflection in Action. Manning Publications. 2005.
10. Fowler M., Parsons R. Domain-Specific Languages. Addison-Wesley. 2010.
11. Fowler M., Rice D., Foemmel M., Hieatt E., Mee R., Stafford R. Patterns of Enterprise Application Architecture. Addison-Wesley. 2002.

12. Goldman N.M. Ontology-oriented programming: Static typing for the inconsistent programmer, in 2nd Int. Conf. Semantic Web. Sanibel Island. 2003.
13. Hazzard K., Bock J. Metaprogramming in .NET. Manning Publications. 2012.
14. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edn. Prentice Hall. 2004.
15. Martin R.C. Agile Software Development: Principles, Patterns and Practices. Pearson Education. 2002.
16. Pan J.Z., Staab S., Aßmann U., Ebert J., Zhao Y. Ontology-Driven Software Development. Springer-Verlag. 2013.
17. Paulheim H. Ontology-based Application Integration. Springer-Verlag. 2011.
18. Smith B.C. Procedural Reflection in Programming Languages, PhD Thesis. MIT. 1982.
19. Stennikov V.A., Barakhtenko E.A., Sokolov D.V., Oshchepkova T.B. Problems of modeling and optimization of heat supply systems: new methods and software for optimization of heat supply system parameters, in Sustaining Power Resources through Energy Optimization and Engineering Premier Reference Source. IGI Global. 2016. Pp. 76–101.
20. Stennikov V.A., Barakhtenko E.A., Sokolov D.V. Use of Multilevel Modeling for Determining Optimal Parameters of Heat Supply Systems // Thermal Engineering. 2017. №64. Pp. 518–525.
21. Štuikys V., Damaševičius R. Meta-Programming and Model-Driven Meta-Program Development. Springer-Verlag. 2013.
22. Volter M., Stahl T., Bettin J., Haase A., Helsen S. Model-Driven Software Development: Technology, Engineering, Management. Wiley. 2006.

UDK 004.415.2

AUTOMATION OF THE INTEGRATED GRAPHICAL ENVIRONMENT CONSTRUCTION FOR COMPUTER MODELING OF ENERGY PIPELINE SYSTEMS

Dmitry V. Sokolov
PhD., Senior Researcher, e-mail: sokolov_dv@isem.irk.ru
Evgeny A. Barakhtenko
PhD., Senior Researcher, e-mail: barakhtenko@isem.irk.ru
Melentiev Energy Systems Institute
Siberian Branch of the Russian Academy of Sciences
130, Lermontov Str., 664033, Irkutsk, Russia

Abstract. The paper presents a methodological approach to automated construction of integrated graphical environment for computer modeling of pipeline systems of various types. Automated construction of the integrated environment is performed using a computer model of the pipeline system and ontologies based on the Model Driven Engineering concept and reflection programming.

Keywords: integrated graphical environment, automation of programming, Model-Driven Engineering, metaprogramming, reflective programming, ontology.

References

1. Vorozhtsova T.N., Skripkin S.K. Ispol'zovaniye ontologiy pri modelirovanii programmnogo kompleksa [The use of ontologies in modeling software] // Vychislitel'nye tekhnologii = Computational Technologies. 2008. vol.13. part I. Pp. 376–381. (in Russian)
2. Vorozhtsova T.N., Skripkin S.K. [The ontological approach to software modeling] // Vestnik IrGTU = Proceedings of Irkutsk State Technical University. 2006. No. 2(26). Pp. 72–78. (in Russian)
3. Stennikov V.A., Barakhtenko E.A., Sokolov D.V. Razrabotka printsipov postroyeniya integrirovannoy graficheskoy sredy dlya komp'yuternogo modelirovaniya truboprovodnykh sistem energetiki [Development of principles for the creation of an integrated graphical environment for computer modeling of energy pipeline systems] // Informatsionnyye tekhnologii = Information technologies. 2018. vol. 24. Pp. 313–320. (in Russian)
4. Beck K. Extreme Programming Explained: Embrace Change. Addison-Wesley. 1999.
5. Booch G. Object-Oriented Analysis and Design with Applications. Addison-Wesley. 2007.
6. Booch G., Jacobson I., Rumbaugh J. The Unified Software Development Process. Prentice Hall. 1999.
7. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide, 2nd edn. Addison-Wesley. 2005.
8. Brambilla M., Cabot J., Wimmer M. Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering. Morgan & Claypool. 2012.
9. Forman I., Forman N. Java Reflection in Action. Manning Publications. 2005.
10. Fowler M., Parsons R. Domain-Specific Languages. Addison-Wesley. 2010.
11. Fowler M., Rice D., Foemmel M., Hieatt E., Mee R., Stafford R. Patterns of Enterprise Application Architecture. Addison-Wesley. 2002.
12. Goldman N.M. Ontology-oriented programming: Static typing for the inconsistent programmer, in 2nd Int. Conf. Semantic Web. Sanibel Island. 2003.
13. Hazzard K., Bock J. Metaprogramming in .NET. Manning Publications. 2012.
14. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edn. Prentice Hall. 2004.
15. Martin R.C. Agile Software Development: Principles, Patterns and Practices. Pearson Education. 2002.
16. Pan J.Z., Staab S., Aßmann U., Ebert J., Zhao Y. Ontology-Driven Software Development. Springer-Verlag. 2013.
17. Paulheim H. Ontology-based Application Integration. Springer-Verlag. 2011.
18. Smith B.C. Procedural Reflection in Programming Languages, PhD Thesis. MIT. 1982.
19. Stennikov V.A., Barakhtenko E.A., Sokolov D.V., Oshchepkova T.B. Problems of modeling and optimization of heat supply systems: new methods and software for optimization of heat supply system parameters, in Sustaining Power Resources through Energy Optimization and Engineering Premier Reference Source. IGI Global. 2016. Pp. 76–101.
20. Stennikov V.A., Barakhtenko E.A., Sokolov D.V. Use of Multilevel Modeling for Determining Optimal Parameters of Heat Supply Systems // Thermal Engineering. 2017. №64. Pp. 518–525.
21. Štuikys V., Damaševičius R. Meta-Programming and Model-Driven Meta-Program Development. Springer-Verlag. 2013.
22. Volter M., Stahl T., Bettin J., Haase A., Helsen S. Model-Driven Software Development: Technology, Engineering, Management. Wiley. 2006.