УДК 004.415.2

МЕТОДИЧЕСКИЙ ПОДХОД К РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОЕКТИРОВАНИЯ СИСТЕМ ТЕПЛОСНАБЖЕНИЯ

Барахтенко Евгений Алексеевич,
К.т.н., с.н.с., e-mail: barakhtenko@isem.irk.ru
Соколов Дмитрий Витальевич,
К.т.н., с.н.с., e-mail: sokolov_dv@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 новый методический подход к автоматизированному построению программного обеспечения для решения задач проектирования систем теплоснабжения. Методический подход основан на парадигме Model-Driven Engineering. Суть этой парадигмы заключается в том, что программное обеспечение создается на основе моделей. Знания о системах теплоснабжения, прикладных задачах и прикладном программном обеспечении сохраняются для многократного использования в виде онтологий. Автоматизированное построение программного комплекса осуществляется на основе компьютерной модели системы теплоснабжения, онтологий и современных технологий метапрограммирования. Предложенный подход позволяет успешно решить проблему разделения методов решения прикладных задач и математических моделей элементов системы теплоснабжения. С этой целью методы реализуются в виде программных компонентов, которые не связаны со свойствами и моделями конкретного оборудования. А модели элементов системы теплоснабжения автоматически компилируются в программные компоненты. В процессе построения программной системы программные компоненты, реализующие модели и методы, динамически интегрируются в создаваемую систему. В результате программная система, ориентированная на решение конкретной прикладной задачи, создается в автоматизированном режиме. Разработанный подход использован для реализации программного комплекса СОСНА, который применяется для проектирования городских систем теплоснабжения.

Ключевые слова: методический подход, Model-Driven Engineering, онтология, метапрограммирование, проектирование систем теплоснабжени.

Цитирование: Барахтенко Е.А., Соколов Д.В. Методический подход к разработке программного обеспечения для проектирования систем теплоснабжения // Информационные и математические технологии в науке и управлении. 2019. № 3 (15). С. 53–65. DOI: 10.25729/2413-0133-2019-3-05

Введение. В ИСЭМ СО РАН в качестве теоретической основы для решения задач проектирования и управления функционированием трубопроводных систем предложена теория гидравлических цепей (ТГЦ). Она служит базой для моделирования, расчета, оценивания и оптимизации трубопроводных и гидравлических систем различного типа и

назначения [4]. В рамках ТГЦ выделяется проблема оптимального проектирования систем теплоснабжения (ТСС), которая охватывает широкий круг задач и состоит в поиске оптимального направления изменения структуры и параметров систем, определения и устранения «узких» мест, замены устаревших технологий и оборудования на новые энергоэффективные решения, обеспечения требований надежности теплоснабжения и управляемости систем при удовлетворении физико-технических условий их функционирования и выполнении ограничений на режимные параметры [7]. Особенность решения проблемы оптимального проектирования ТСС состоит в том, что она предполагает формирование своего алгоритма из различного состава подзадач (выбор структуры, определение оптимальных параметров, анализ надежности системы, тепловые и гидравлические расчеты, расчет параметров источников тепла и др.), индивидуального для множества рассматриваемых ТСС, с учетом их особенностей. Как правило, это сложный итерационный вычислительный процесс, в ходе которого подзадачи для различных ТСС могут решаться в разной последовательности и различными методами в зависимости от поставленной цели. Один из возможных алгоритмов решения задачи представлен на рис. 1 [7].

Рис. 1. Алгоритм решения задачи проектирования систем теплоснабжения

Методические подходы, традиционно применяемые при разработке программного обеспечения для проектирования ТСС, не позволяют реализовать разработанную в рамках ТГЦ методику решения этих задач, которая требует применения гибких схем организации

вычислительного процесса, замены элементов программной системы, учета особенностей развития конкретной ТСС и используемого для ее построения широкого спектра энергетического оборудования, многократного использования реализованного программного обеспечения и его настройки на особенности конкретного набора используемого оборудования.

Существующие подходы к разработке программного обеспечения представлены в [8, 10, 25]. Принципы разработки универсальных программных компонентов рассмотрены в [25]. В [16, 17] основное внимание уделяется общим принципам разработки сложных программных систем. Авторы [13, 18, 28, 29, 35, 36] предлагают парадигму модельно-ориентированного проектирования (MDE) для автоматизации этапов разработки программного обеспечения. Парадигма MDE представляет собой набор методических подходов к автоматизированному проектированию сложных программных систем на основе предварительно разработанных моделей. В [11, 12, 24] язык UML предлагается в качестве универсального инструмента для описания предметных областей. В настоящее время существуют подходы, которые предполагают использование онтологий в качестве инструмента для описания предметной области [19, 20, 26, 27]. Также онтологии используются для разработки программного обеспечения [1 - 3, 9, 14, 23]. Авторы [15, 22, 35] рассматривают подходы, предназначенные для автоматизации этапов разработки программного обеспечения на основе метапрограммирования. Рефлексивное программирование является одним из типов метапрограммирования. Он представляет собой расширение парадигмы объектно-ориентированного программирования [15, 22, 30]. Рефлексивное программирование позволяет проверять структуру программной системы и динамически изменять набор программных компонентов в процессе ее работы.

В настоящей статье излагается разработанный авторами методический подход к автоматизированному построению программного обеспечения для решения задач проектирования ТСС. Этот подход основан на применении парадигмы MDE. При автоматизированном построении программного обеспечения применяются современные технологии метапрограммирования и онтологии, которые позволяют формализовано описать объекты предметной области, их свойства и взаимосвязи между этими объектами [21, 31, 32].

1. Постановка задач исследования. Одна из особенностей предметной области проектирования ТСС состоит в том, что реализация программного обеспечения, предназначенного для решения этих задач, является завершающим этапом разработки методов, математических моделей элементов ТСС, методик и алгоритмов. Применение этого программного обеспечения при решении научных и практических инженерных задач приводит к накоплению опыта, который позволяет разрабатывать более точные математические модели, уточнять справочную информацию, повышать быстродействие алгоритмов, улучшать сходимость методов, получать оригинальное решение какой-либо практической задачи. Как правило, накопленный опыт фиксируется путем внесения изменений в программное обеспечение, что повышает его качество и соответствие реальным инженерным системам. Описанный подход к разработке приводит к тому, что программное обеспечение становится единственным средством хранения всего накопленного опыта. В результате этот опыт недоступен для изучения и использования широкому кругу специалистов.

В методических подходах, традиционно применяемых при построении программного обеспечения, отсутствует четкое разделение на методы (алгоритмы, методики) решения прикладных задач и математические модели элементов ТСС. В результате программные модули, реализующие алгоритмы, становятся ориентированными на конкретные классы задач и набор оборудования, что значительно затрудняет их настройку под конкретную решаемую задачу и многократное использование при построении различных программных систем. Отсутствует возможность для исследователя создавать свои модели элементов и интегрировать их в программное обеспечение при проведении научных или инженерных расчетов. Происходит многократное дублирование программной реализации одной и той же модели элемента ТСС в различных вычислительных модулях, поэтому в случае корректировки модели необходимо вносить изменения во все программные модули.

Схема взаимодействия между вычислительными модулями скрыта в программном коде управляющего модуля программной системы, что значительно затрудняет понимание и развитие алгоритма решения задачи. При возникновении необходимости любого преобразования управляющего алгоритма программной системы требуется внесение изменений в существующий управляющий модуль или создание нового. Наличие перечисленных недостатков обусловливает невозможность гибкой организации управления вычислительным процессом, необходимой при реализации методик решения задач оптимального проектирования ТСС.

Для преодоления перечисленных трудностей необходимо разработать новый методический подход к построению программного обеспечения, который позволит в автоматизированном режиме создавать сложные программные системы, ориентированные на решение прикладных задач с учетом индивидуальных особенностей моделируемых ТСС. В рамках этого подхода знания о предметной области должны быть формализованы в виде онтологий, что позволит многократно использовать их при автоматическом построении прикладных программных систем.

2. Предложенный методический подход. В результате проведенных авторами исследований предложен новый методический подход к разработке программного обеспечения на базе парадигмы MDE, который позволяет успешно преодолевать перечисленные ранее проблемы. В рамках этого подхода MDE адаптирована к особенностям методического и программного обеспечения, применяемого при решении задач проектирования ТСС: программная система строится в автоматизированном режиме на основе компьютерной модели конкретной ТСС, предварительно разработанных моделей элементов ТСС, программных компонентов, реализующих методы и алгоритмы решения прикладных задач, и знаний, формализованных в виде системы онтологий.

Предложенный подход позволяет успешно решить задачу разделения методов решения прикладных задач и моделей элементов ТСС. Для этого методы реализуются в виде программных компонентов, которые не привязаны к свойствам и моделям конкретного оборудования. Модели элементов ТСС автоматически компилируются в программные компоненты. В процессе построения программной системы выполняется динамическая интеграция программных компонентов, реализующих модели элементов ТСС и методы решения задач, что позволяет создать программную систему, ориентированную на решение конкретной прикладной задачи.

Разработанный методический подход включает:

- Методику автоматизированного построения сложных программных систем на основе применения парадигмы MDE, современных методов метапрограммирования и онтологий.
- Принципы построения системы онтологий для хранения знаний о ТСС, связанных с этими системами задачах и используемом программном обеспечении.
- Принципы построения расширяемой архитектуры программного обеспечения, ориентированного на особенности решаемой прикладной задачи.

Основные особенности разработанного методического подхода:

- методический подход ориентирован на разработку программного обеспечения для решения задач оптимального проектирования ТСС;
- модель программной системы строится автоматически на основе компьютерной модели ТСС, описания прикладной задачи и знаний, хранение которых организовано в виде системы онтологий;
- алгоритм верхнего уровня, определяющий схему взаимодействия программных компонентов и ход вычислительного процесса, строится автоматически во время построения программной системы на основе описания прикладной задачи и методики ее решения;
- привлечение современных технологий метапрограммирования предоставляет возможность автоматизированного формирования программной системы и гибкой настройки ее на особенности развития и состав оборудования конкретной ТСС.

Для автоматизированного построения программного обеспечения и информационного наполнения его пользовательского интерфейса используется система онтологий [6]. Эта система состоит из следующих онтологий.

Метаонтология. Содержит базовые понятия, которые используются при построении прикладных онтологий.

Онтология ТСС. Содержит описание элементов системы и их свойств.

Онтология задач. Содержит описание прикладных задач, методов и алгоритмов их решения, параметров, являющихся исходными данными, и параметров, получаемых в результате решения задачи.

Онтология ПО. Содержит описание программных компонентов и их свойств, метаданные (входные и выходные параметры, описание форматов данных) и описание технологий и интерфейсов доступа к программным компонентам.

В качестве средства формального описания онтологий в рамках предложенного методического подхода используется предметно-ориентированный язык на базе XML. Для представления графических и математических моделей используются соответственно языки SVG и MathML, являющиеся подмножествами языка XML.

Предложенный методический подход позволяет в автоматизированном режиме создавать сложные программные системы для решения задач проектирования ТСС и решать прикладные задачи, используя разработанные в рамках ТГЦ методики, методы и алгоритмы.

3. Методика автоматизированного построения программной системы. Предложенная авторами методика автоматизированного построения программной системы состоит из следующих этапов [5].

Этап 1. Построение компьютерной модели конкретной ТСС.

На этом этапе инженером создается компьютерная модель конкретной ТСС, отражающая свойства сети: структуру, набор используемого оборудования и его свойства, параметры элементов системы (технические, гидравлические, граничные условия) [4]. Эта модель сохраняется в базе данных для многократного использования.

Этап 2. Формализация прикладной задачи.

На этом этапе происходит формальное описание прикладной задачи: задаются параметры задачи математического моделирования и оптимизации ТСС, задается набор допустимого к установке оборудования, задаются ограничения и условия решения прикладной задачи.

Этап 3. Автоматическое построение модели программной системы

На этом этапе автоматически создается модель программной системы, ориентированной на решение прикладной задачи (рис. 2). Эта модель представляет собой совокупность структур данных, описывающих свойства и структуру создаваемой прикладной системы.

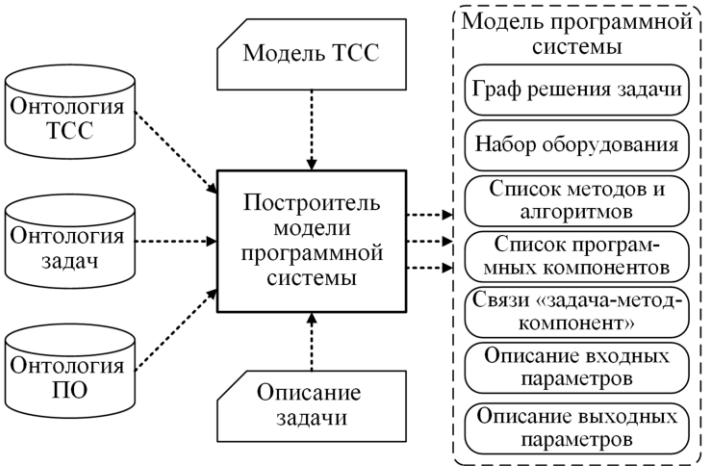

Рис. 2. Схема построения модели программной системы

Этап 4. Автоматическое построение программной системы на основе ее модели при помощи применения технологий метапрограммирования. В рамках методики используются следующие технологии: 1) автоматическая генерация программного кода; 2) динамическая компиляция программного кода; 3) рефлексивное программирование. Путем генерации программного кода и его компиляции создаются программные реализации моделей элементов ТСС. При помощи технологий рефлексивного программирования компоненты подключаются к программной системе и настраиваются на решение прикладной задачи.

Этап 5. Применение программной системы при решении прикладных задач. В результате выполнения этапов 1–4 создается программная система, представленная на рис. 3.

Программная система состоит из трех архитектурных слоев: 1) подсистема управления вычислением, которая содержит компонент супервизор, управляющий ходом вычисленного процесса на основе графа решения задачи; 2) подсистема доступа к данным, которая обеспечивает обмен информацией между базами данных (БД) и структурами данных локальной памяти при помощи загрузчика данных и рефлексивного компонента; 3) вычислительная подсистема, которая решает прикладную задачу путем использования

программных компонентов, реализующих методы и модели. Предложенная архитектура обеспечивает необходимую гибкость и адаптацию к различному составу оборудования.

4. Программный комплекс СОСНА. Предлагаемый авторами методический подход успешно применяется в ИСЭМ СО РАН при разработке программного обеспечения для решения задач проектирования ТСС. На основе этого подхода реализован программный комплекс (ПК) нового поколения СОСНА (Синтез Оптимальных Систем с учетом НАдежности) [7], предназначенный для решения сложной инженерной задачи выбора оптимальных параметров ТСС, многократно решаемой при их проектировании [4, 33, 34].

ПК СОСНА представляет собой сложную прикладную программную систему, генерируемую в автоматизированном режиме в контексте решения прикладной задачи на основе компьютерной модели исследуемой ТСС, описания решаемой задачи и знаний о предметной области. Хранение знаний о предметной области организовано в виде системы онтологий. В процессе автоматизированного построения программной системы средствами метапрограммирования используется разработанная авторами библиотека программных компонентов. Получаемая в результате система имеет архитектуру, представленную на рис. 3.

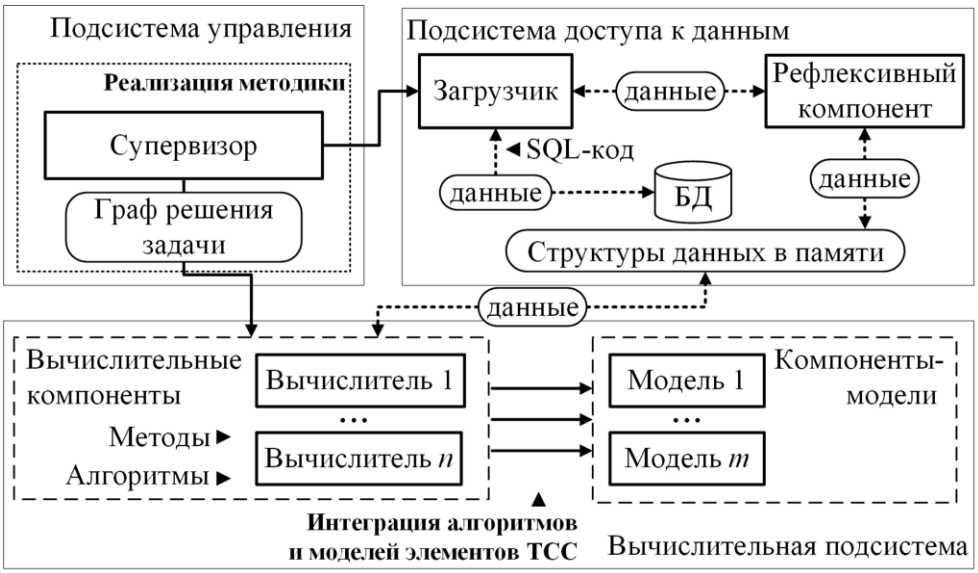

Рис. 3. Архитектура программной системы, получаемой в результате автоматического построения

С помощью ПК СОСНА проводились многовариантные расчеты достаточно сложных ТСС для решения задач их проектирования, а также выполнены научные исследования разработанных методов и алгоритмов. С помощью ПК выполнены расчеты ТСС Центрального и Адмиралтейского районов Санкт-Петербурга, города Братска и поселка Магистральный. На рис. 4 приведены полученные по результатам расчетов целесообразные мероприятия по реконструкции ТСС г. Братска.

Результаты проведенных расчетов были использованы при подготовке рекомендаций по оптимальной реконструкции перечисленных выше и других ТСС.

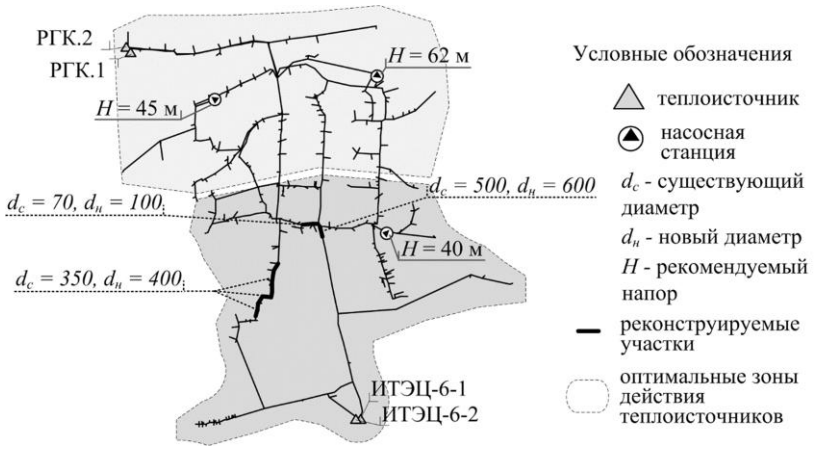

Рис. 4. Схема теплоснабжающей системы г. Братска, с обозначенными рекомендованными мероприятиями по ее реконструкции

Заключение. Предложен методический подход к разработке программного обеспечения, основанный на применении парадигмы MDE, который позволяет в автоматизированном режиме создавать сложные программные системы для решения научных и прикладных задач оптимального проектирования сложных технических систем. Применение разработанного методического подхода при реализации ПК СОСНА обеспечило создание открытой программной системы, приспособленной для развития и сопровождения. Этот подход может применяться в научно-исследовательских и проектных организациях, занимающихся вопросами разработки программного обеспечения для проектирования ТСС. Практическое применение разработанного методического и программного обеспечения позволяет получать рекомендации по преобразованию ТСС, повышающие эффективность их работы и качество снабжения потребителей тепловой энергией.

Исследование выполнено в ИСЭМ СО РАН при поддержке Российского научного фонда (грант № 17-19-01209).

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. Загорулько Ю.А., Загорулько Г.Б. Онтологический подход к разработке системы поддержки принятия решении на нефтегазодобывающем предприятии // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2012. Т. 10. № 1. С. 121–128.
4. Меренков А.П., Хасилев В.Я. Теория гидравлических цепей. М.: Наука. 1985. 280 с.
5. Стенников В.А., Барахтенко Е.А., Соколов Д.В. Применение концепции Model-Driven Engineering в программном комплексе для определения оптимальных параметров теплоснабжающих систем // Программная инженерия. 2016. Т. 7. № 3. С. 108–116.
6. Стенников В.А., Барахтенко Е.А., Соколов Д.В. Применение онтологий при реализации концепции модельно-управляемой разработки программного обеспечения для

проектирования теплоснабжающих систем // Онтология проектирования. №4(14). 2014. С. 54–68.
7. Стенников В.А., Сеннова Е.В., Ощепкова Т.Б. Методы комплексной оптимизации развития теплоснабжающих систем // Энергетика. М.: Изв. РАН. 2006. №3. С. 44–54.
8. Beck K. Extreme Programming Explained: Embrace Change. Boston: Addison-Wesley. 1999.
9. Berman A.F., Nikolaychuk O.A., Pavlov A.I. The Ontology Model for Automating the Solution of Multidisciplinary Research Tasks // Proc. the V Intern. Workshop Critical Infrastructures: Contingency Management. Intelligent. Agent-Based. Cloud Computing and Cyber Security (IWCI 2018). 2018. Vol. 158. Pp. 1–6.
10. Booch G. Object-Oriented Analysis and Design with Applications. Boston: Addison-Wesley. 2007.
11. Booch G., Jacobson I., Rumbaugh J. The Unified Software Development Process. Upper Saddle River, New Jersey: Prentice Hall. 1999.
12. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide, 2nd edn. Boston: Addison-Wesley. 2005.
13. Brambilla M., Cabot J., Wimmer M. Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering. San Rafael: Morgan & Claypool. 2012.
14. Chungoora N., Young R.I., Gunendran G., Palmer C., Usman Z., Anium N.A., Cutting-Decellec A-F., Harding J.A., Case K. A model-driven ontology approach for manufacturing system interoperability and knowledge sharing // Computers in Industry. 2013. Vol. 64. no 4. Pp. 392–401.
15. Forman I., Forman N. Java computection in Action. Greenwich: Manning Publications. 2005.
16. Fowler M., Parsons R. Domain-Specific Languages. Boston: Addison-Wesley. 2010.
17. Fowler M., Rice D., Foemmel M., Hieatt E., Mee R., Stafford R. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley. 2002.
18. Gascuena J.M., Navarro E., Fernandez-Caballero A. Model-driven engineering techniques for the development of multi-agent systems // Engineering Applications of Artificial Intelligence. 2012. Vol. 25. no 1. Pp. 159–173.
19. Gavrilova T., Leshcheva I. The interplay of knowledge engineering and cognitive psychology: learning ontologies creating // International Journal of Knowledge and Learning. 2015. Vol. 10. no 2. Pp. 182–197.
20. Goldman N.M. Ontology-Oriented Programming: Static Typing for the Inconsistent Programmer // The Semantic Web - ISWC 2003. ISWC 2003. Lecture Notes in Computer Science. 2003. Vol. 2870.
21. Gruber T.R. A translation approach to portable ontology specifications // Knowledge Acquisition. 1993. Vol. 5. no 2. Pp. 199–220.
22. Hazzard K., Bock J. Metaprogramming in .NET. Greenwich: Manning Publications. 2012.
23. Katasonov A. Ontology-driven software engineering: beyond model checking and transformations // International Journal of Semantic Computing. 2012. Vol. 6. no 2. Pp. 205–242.
24. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edn. Upper Saddle River, New Jersey: Prentice Hall. 2004.

25. Martin R.C. Agile Software Development: Principles, Patterns and Practices. New York: Pearson Education. 2002.
26. Pan J.Z., Staab S., Aßmann U., Ebert J., Zhao Y. Ontology-Driven Software Development. Berlin: Springer-Verlag. 2013.
27. Paulheim H. Ontology-based Application Integration. New York: Springer-Verlag. 2011.
28. Schmidt D.C. Guest Editor's Introduction: Model-Driven Engineering // Computer. 2006. Vol. 39. no 2. Pp. 25–31.
29. Silva A.R. Model-driven engineering: A survey supported by the unified conceptual model // Computer Languages, Systems & Structures. 2015. Vol. 43. Pp. 139–155.
30. Smith B.C. Procedural Reflection in Programming Languages, PhD Thesis. Massachusetts: MIT. 1982.
31. Staab S., Studer R. Handbook on Ontologies, 2nd edn. Heidelberg: Springer-Verlag. 2009.
32. Staab S., Walter T., Gröner G., Parreiras F.S. Model Driven Engineering with Ontology Technologies // Reasoning Web. Semantic Technologies for Software Engineering. Lecture Notes in Computer Science. 2010. Vol. 6325. Pp. 62–98.
33. Stennikov V., Barakhtenko E., Sokolov D. Use of multilevel modeling for determining optimal parameters of heat supply systems // Thermal Engineering. 2017. Vol. 64. Pp. 518–525.
34. Stennikov V., Barakhtenko E., Sokolov D., Oshchepkova T. Problems of modeling and optimization of heat supply systems: new methods and software for optimization of heat supply system parameters // Sustaining power resources through energy optimization and engineering premier reference source. USA: IGI Global. 2016. Pp. 76–101.
35. Štuikys V., Damaševičius R. Meta-Programming and Model-Driven Meta-Program Development. London: Springer-Verlag. 2013.
36. Volter M., Stahl T., Bettin J., Haase A., Helsen S. Model-Driven Software Development: Technology, Engineering, Management. New York: Wiley. 2006.

UDK 004.415.2

AN APPROACH TO THE SOFTWARE DEVELOPMENT FOR THE DESIGN OF HEAT SUPPLY SYSTEMS

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

Abstract. The paper presents a new approach to the automated construction of software for solving design problems of heating systems. The approach is based on the Model-Driven Engineering paradigm. The essence of this paradigm lies in the fact that software is generated on the basis of a formal description presented by models. Knowledge of heat supply systems, applied tasks and applied software is formalized in the form of

ontologies. The automated construction of a software is carried out on the basis of a computer model of a heat supply system, ontologies and metaprogramming technologies. The proposed approach allows us to successfully solve the problem of separation of methods for solving applied tasks and models of heat supply system elements. To this end, the methods are implemented in the form of software components that are not related to the properties and models of specific equipment. And the models of heat supply system elements are automatically compiled into software components. In the process of building a software system, software components that implement models and methods are dynamically integrated. As a result, a software system oriented to the solution of a specific applied task is created in an automated mode. The developed approach was applied for the implementation of the SOSNA software. SOSNA is used to design urban heat supply systems.

Keywords: approach to the software development, Model-Driven Engineering, ontology, metaprogramming, design of heat supply systems.

References

1. Vorozhtsova T.N., Skripkin S.K. Ispol'zovanie ontologii pri modelirovanii programmnogo kompleksa [The use of ontologies in modeling software] // Vychislitel'nye tekhnologii = Computing Technology. 2008. vol. 13. part I. Pp. 376-381. (in Russian)
2. Vorozhtsova T.N., Skripkin S.K. Ontologicheskii podkhod k modelirovaniyu programmnogo kompleksa [The ontological approach to software modeling] // Vestnik IrGTU = ISTU Bulletin. 2006. No. 2(26). Pp. 72-78. (in Russian)
3. Zagorul'ko Yu.A., Zagorul'ko G.B. Ontologicheskii podkhod k razrabotke sistemy podderzhki prinyatiya reshenii na neftegazodobyvayushchem predpriyatii [Ontological approach to the development of a decision support system at an oil and gas company] // Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya: Informatsionnye tekhnologii = Bulletin of Novosibirsk State University. Series: Information Technology. 2012. vol. 10. no. 1. Pp. 121-128. (in Russian)
4. Merenkov A.P., Khasilev V.Ya. Teoriya gidravlicheskikh tsepey [The theory of hydraulic circuits]. Moscow. Nauka = Science, 1985. 280 p. (in Russian)
5. Stennikov V.A., Barakhtenko E.A., Sokolov D.V. Primenenie kontseptsii Model-Driven Engineering v programmnom komplekse dlya opredeleniya optimal'nykh parametrov teplosnabzhayushchikh sistem [Model-Driven Engineering in the software for parameter optimization of heat supply systems] // Programmnaya inzheneriya = Software engineering. 2016. vol. 7. no 3. Pp. 108-116. (in Russian)
6. Stennikov V.A., Barakhtenko E.A., Sokolov D.V. Primenenie ontologii pri realizatsii kontseptsii model'no-upravlyaemoi razrabotki programmnogo obespecheniya dlya proektirovaniya teplosnabzhayushchikh sistem [Using model-driven engineering and ontologies for the development of a software for heat supply system design] // Ontologiya proektirovaniya = Ontology of Designing. 2014. no. 4(14). Pp. 54-68. (in Russian).
7. Stennikov V.A., Sennova E.V., Oshchepkova T.B. Metody kompleksnoi optimizatsii razvitiya teplosnabzhayushchikh sistem [Methods of complex optimization for heat supply system development] // Energetika = Energy. 2006. no. 3. Pp. 44-54. (in Russian)
8. Beck K. Extreme Programming Explained: Embrace Change. Boston: Addison-Wesley. 1999.

9. Berman A.F., Nikolaychuk O.A., Pavlov A.I. The Ontology Model for Automating the Solution of Multidisciplinary Research Tasks // Proc. the V Intern. Workshop Critical Infrastructures: Contingency Management. Intelligent. Agent-Based. Cloud Computing and Cyber Security (IWCI 2018). 2018. Vol. 158. Pp. 1–6.
10. Booch G. Object-Oriented Analysis and Design with Applications. Boston: Addison-Wesley. 2007.
11. Booch G., Jacobson I., Rumbaugh J. The Unified Software Development Process. Upper Saddle River, New Jersey: Prentice Hall. 1999.
12. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide, 2nd edn. Boston: Addison-Wesley. 2005.
13. Brambilla M., Cabot J., Wimmer M. Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering. San Rafael: Morgan & Claypool. 2012.
14. Chungoora N., Young R.I., Gunendran G., Palmer C., Usman Z., Anium N.A., Cutting-Decellec A-F., Harding J.A., Case K. A model-driven ontology approach for manufacturing system interoperability and knowledge sharing // Computers in Industry. 2013. Vol. 64. no 4. Pp. 392–401.
15. Forman I., Forman N. Java computection in Action. Greenwich: Manning Publications. 2005.
16. Fowler M., Parsons R. Domain-Specific Languages. Boston: Addison-Wesley. 2010.
17. Fowler M., Rice D., Foemmel M., Hieatt E., Mee R., Stafford R. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley. 2002.
18. Gascuena J.M., Navarro E., Fernandez-Caballero A. Model-driven engineering techniques for the development of multi-agent systems // Engineering Applications of Artificial Intelligence. 2012. Vol. 25. no 1. Pp. 159–173.
19. Gavrilova T., Leshcheva I. The interplay of knowledge engineering and cognitive psychology: learning ontologies creating // International Journal of Knowledge and Learning. 2015. Vol. 10. no 2. Pp. 182–197.
20. Goldman N.M. Ontology-Oriented Programming: Static Typing for the Inconsistent Programmer // The Semantic Web - ISWC 2003. ISWC 2003. Lecture Notes in Computer Science. 2003. Vol. 2870.
21. Gruber T.R. A translation approach to portable ontology specifications // Knowledge Acquisition. 1993. Vol. 5. no 2. Pp. 199–220.
22. Hazzard K., Bock J. Metaprogramming in .NET. Greenwich: Manning Publications. 2012.
23. Katasonov A. Ontology-driven software engineering: beyond model checking and transformations // International Journal of Semantic Computing. 2012. Vol. 6. no 2. Pp. 205–242.
24. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edn. Upper Saddle River, New Jersey: Prentice Hall. 2004.
25. Martin R.C. Agile Software Development: Principles, Patterns and Practices. New York: Pearson Education. 2002.
26. Pan J.Z., Staab S., Aßmann U., Ebert J., Zhao Y. Ontology-Driven Software Development. Berlin: Springer-Verlag. 2013.
27. Paulheim H. Ontology-based Application Integration. New York: Springer-Verlag. 2011.

28. Schmidt D.C. Guest Editor's Introduction: Model-Driven Engineering // Computer. 2006. Vol. 39. no 2. Pp. 25–31.
29. Silva A.R. Model-driven engineering: A survey supported by the unified conceptual model // Computer Languages, Systems & Structures. 2015. Vol. 43. Pp. 139–155.
30. Smith B.C. Procedural Reflection in Programming Languages, PhD Thesis. Massachusetts: MIT. 1982.
31. Staab S., Studer R. Handbook on Ontologies, 2nd edn. Heidelberg: Springer-Verlag. 2009.
32. Staab S., Walter T., Gröner G., Parreiras F.S. Model Driven Engineering with Ontology Technologies // Reasoning Web. Semantic Technologies for Software Engineering. Lecture Notes in Computer Science. 2010. Vol. 6325. Pp. 62–98.
33. Stennikov V., Barakhtenko E., Sokolov D. Use of multilevel modeling for determining optimal parameters of heat supply systems // Thermal Engineering. 2017. Vol. 64. Pp. 518–525.
34. Stennikov V., Barakhtenko E., Sokolov D., Oshchepkova T. Problems of modeling and optimization of heat supply systems: new methods and software for optimization of heat supply system parameters // Sustaining power resources through energy optimization and engineering premier reference source. USA: IGI Global. 2016. Pp. 76–101.
35. Štuikys V., Damaševičius R. Meta-Programming and Model-Driven Meta-Program Development. London: Springer-Verlag. 2013.
36. Volter M., Stahl T., Bettin J., Haase A., Helsen S. Model-Driven Software Development: Technology, Engineering, Management. New York: Wiley. 2006.