

УДК 004.415.2

ПРИМЕНЕНИЕ ОНТОЛОГИЙ ПРИ РЕАЛИЗАЦИИ КОНЦЕПЦИИ МОДЕЛЬНО-УПРАВЛЯЕМОЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОЕКТИРОВАНИЯ ТЕПЛОСНАБЖАЮЩИХ СИСТЕМ

В.А. Стенников¹, Е.А. Барахтенко², Д.В. Соколов³

*Институт систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук,
Иркутск, Россия*

¹sva@isem.sei.irk.ru, ²barakhtenko@isem.sei.irk.ru, ³sokolov_dv@isem.sei.irk.ru

Аннотация

В статье излагается разработанный авторами методический подход к построению программного обеспечения для решения задач проектирования теплоснабжающих систем. Этот подход основан на применении концепции модельно-управляемой разработки (Model-Driven Engineering), которая представляет собой совокупность методических принципов автоматического построения сложных программных систем на основе предварительно разработанных моделей. При построении программного обеспечения применяются современные технологии метапрограммирования и формализованные в виде онтологической системы знания предметной области. Излагаются принципиальные положения разработанной в ходе выполненной работы методики автоматического построения программной системы. В статье представлены принципы построения инструментальной платформы, которая служит в качестве единой основы при автоматическом построении программного обеспечения для решения задач оптимального проектирования теплоснабжающих систем. Приведено описание программного комплекса СОСНА, разработанного на базе предложенного методического подхода.

Ключевые слова: *Model-Driven Engineering, метапрограммирование, онтология, программный комплекс, онтологическая система, методический подход, автоматизация программирования, архитектура программного обеспечения, теплоснабжающие системы.*

Введение

Теоретической основой для решения задач проектирования и управления функционированием трубопроводных систем различного типа и назначения является разработанная и успешно развиваемая в Институте систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук (ИСЭМ СО РАН) теория гидравлических цепей (ТГЦ). Она предоставляет методическую базу для моделирования, расчета, оценивания и оптимизации трубопроводных и гидравлических систем различного типа [1]. В рамках ТГЦ выделяется проблема оптимального проектирования теплоснабжающих систем (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 [2]. Особенность решения проблемы оптимального проектирования ТСС состоит в том, что она предполагает формирование каждый раз нового алгоритма из различного состава подзадач (оптимизация структуры, оптимизация параметров, анализ надежности системы, тепловые и гидравлические расчеты, расчет

параметров источников тепла и др.), индивидуального для множества рассматриваемых ТСС и учитывающего их конкретные особенности. 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 рисунке 1 [2].

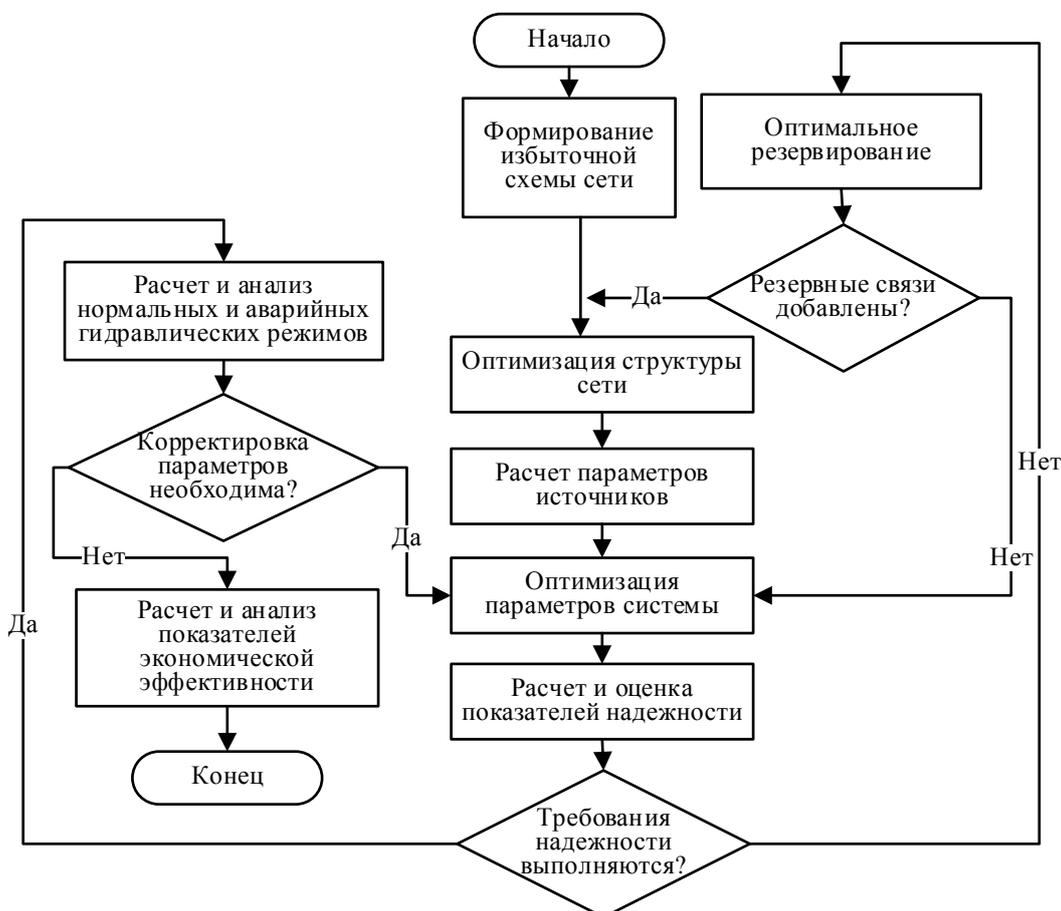


Рисунок 1 – Алгоритм решения задачи проектирования ТСС

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.

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 формализованного 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 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 или создание нового управляющего модуля. Наличие перечисленных недостатков обуславливает невозможность гибкой организации управления вычислительным процессом, необходимого при реализации методик решения задач оптимального проектирования ТСС.

Для преодоления перечисленных трудностей необходимо разработать новый методический подход к построению программного обеспечения, который позволит автоматически 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 [3-5], что позволит многократно их использовать при автоматическом построении прикладных программных систем. Необходимо разработать единую инструментальную платформу, которая позволит автоматически формировать программную систему, ориентированную на решение конкретной прикладной задачи и выполнять сложные инженерные расчеты по поиску оптимальных путей преобразования реальных ТСС с целью повышения их эффективности и надежности функционирования.

1 Новый методический подход

В результате проведенных авторами исследований предложен новый методический подход к разработке программного обеспечения на базе концепции MDE (Model-Driven Engineering, «управляемая моделями разработка») [6-7], который позволяет успешно преодолевать перечисленные проблемы. В рамках этого подхода MDE адаптирована к особенностям методического и программного обеспечения, применяемого при решении задач проектирования ТСС. Программная система автоматически строится на основе следующих компонентов:

- компьютерной модели конкретной ТСС;
- предварительно разработанных моделей элементов ТСС;

- программных компонентов, реализующих методы и алгоритмы решения прикладных задач;
- знаний, формализованных в виде онтологической системы.

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 следующие компоненты.

1. Методику автоматического построения сложных программных систем на основе применения концепции MDE, современных методов метапрограммирования [8] и онтологий.
2. Принципы построения расширяемой архитектуры программного обеспечения, ориентированного на особенности решаемой прикладной задачи.
3. Принципы построения инструментальной платформы как единой универсальной основы для разработки программного обеспечения для решения комплекса задач проектирования ТСС.
4. Методические принципы разработки и применения универсальных программных компонентов.

В процессе автоматического построения программной системы используются онтологии, которые позволяют формализовано описать объекты предметной области, их свойства и взаимосвязи между этими объектами [4]. Онтологии в настоящее время получают все более широкое применение при решении инженерных задач [9]. Существуют работы, в которых описывается опыт применения онтологий при решении задач энергетики [10, 11].

Основные особенности разработанного методического подхода состоят в следующем:

- 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 онтологической системы;
- 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 автоматического формирования программной системы и гибкой настройки ее на особенности развития и состав оборудования конкретной ТСС.

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

2 Методика автоматического построения программной системы

Предложенная авторами методика автоматического построения программной системы включает пять этапов.

Этап 1. Построение компьютерной модели конкретной ТСС.

На этом этапе инженером создается компьютерная модель конкретной ТСС, отражающая ее свойства: структуру, набор используемого оборудования и его свойства, параметры элементов системы (технические, гидравлические, граничные условия) [1]. Эта модель сохраняется в базе данных для многократного использования.

Этап 2. Формализация прикладной задачи.

На этом этапе осуществляется формализованное описание прикладной задачи инженером: 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 (рисунок 2). Эта модель представляет собой совокупность структур данных, описывающих свойства и структуру создаваемой прикладной системы.

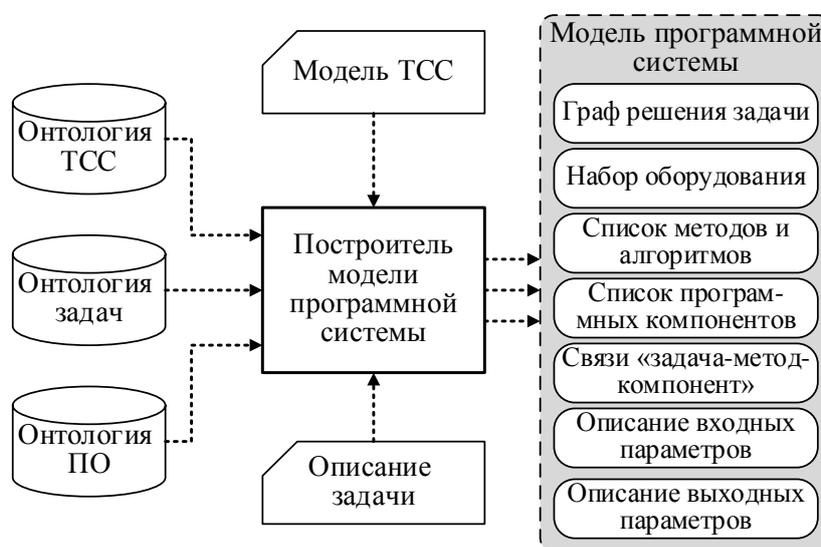


Рисунок 2 – Схема построения модели программной системы

В процессе построения модели программной системы решаются следующие подзадачи.

1. Преобразование компьютерной модели ТСС из базы данных в структуры данных.
2. Создание структур данных, описывающих решаемую прикладную задачу.
3. Создание списка используемого оборудования на основе модели ТСС и описания прикладной задачи.
4. На основе онтологии задач определяется методика решения прикладной задачи, выделяются этапы ее решения, создается список используемых методов и алгоритмов.
5. На основе списка методов и алгоритмов и онтологии программного обеспечения формируется состав необходимых для решения прикладной задачи программных компонентов.
6. На основе описания методики из онтологии задач и описания программных компонентов из онтологии программного обеспечения строится граф решения прикладной задачи, отражающий взаимосвязь и последовательность этапов решения рассматриваемой задачи.
7. Создание структур данных, отражающих связи «задача-метод-компонент».
8. Формирование структур данных, описывающих набор входных и выходных параметров на основе описания методов, алгоритмов и программных компонентов.

Этап 4. Автоматическое построение программной системы на основе ее модели с помощью технологий метапрограммирования [12]. В рамках предлагаемой методики используются следующие технологии:

- автоматическая генерация программного кода;
- динамическая компиляция программного кода;
- рефлексивное программирование.

Схема взаимодействия программных компонентов в процессе построения программной системы представлена на рисунке 3. Посредством генерации программного кода и его компиляции создаются программные реализации моделей элементов тепловой сети. При помощи технологий рефлексивного программирования компоненты подключаются к программной системе и настраиваются на решение прикладной задачи. Рефлексивное программирование представляет собой парадигму программирования, основанную на рефлексии (*англ.* reflection) – подходе к программированию, в соответствии с которым программа может отслеживать и модифицировать собственную структуру и поведение во время выполнения [13].

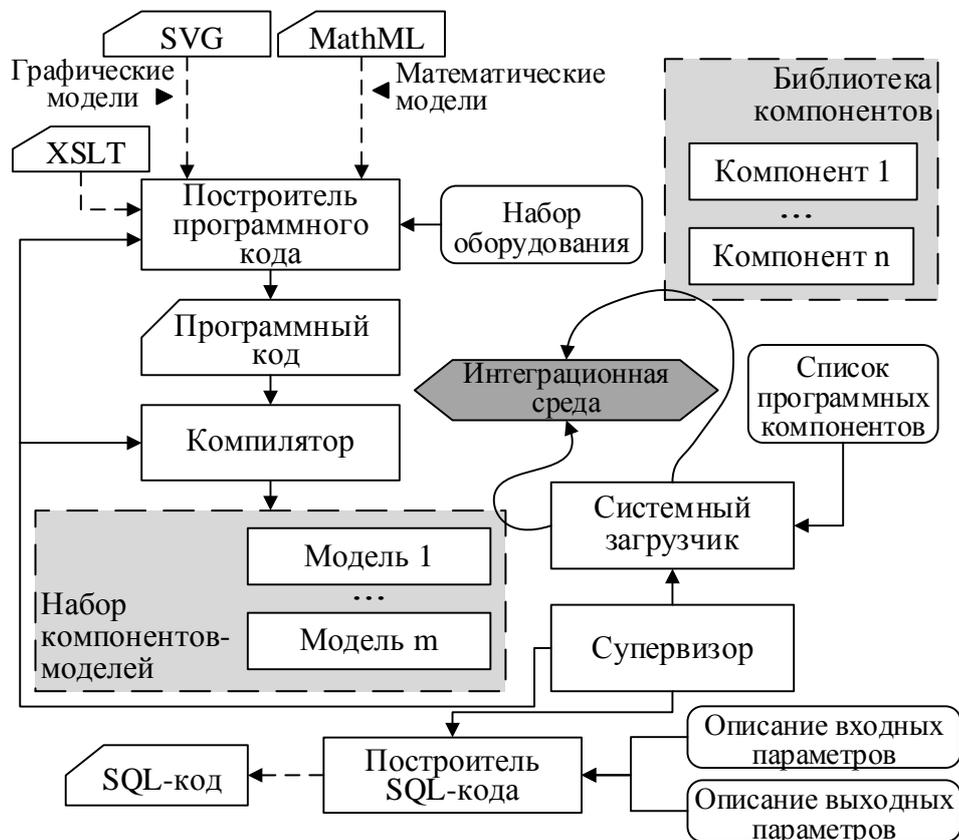


Рисунок 3 – Схема построения программной системы

В рамках этого процесса решаются следующие подзадачи.

1. Преобразование математических и графических моделей элементов ТСС, формализовано описанных в форматах MathML¹ и SVG², в программный код на основе правил, формализованных на языке XSLT³.

¹ <http://www.w3.org/Math/>

² <http://www.w3.org/TR/SVG/>

³ <http://www.w3.org/TR/xslt/>

2. Построение SQL-кода на основе описания входных и выходных параметров.
3. Компиляция программного кода в программные компоненты, реализующие модели элементов ТСС.
4. Загрузка программных компонентов в память. Интеграция этих компонентов в память через стандартизированные интерфейсы, обеспечиваемые паттернами проектирования [14].
5. Формирование структур данных, описывающих используемые программные компоненты.

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

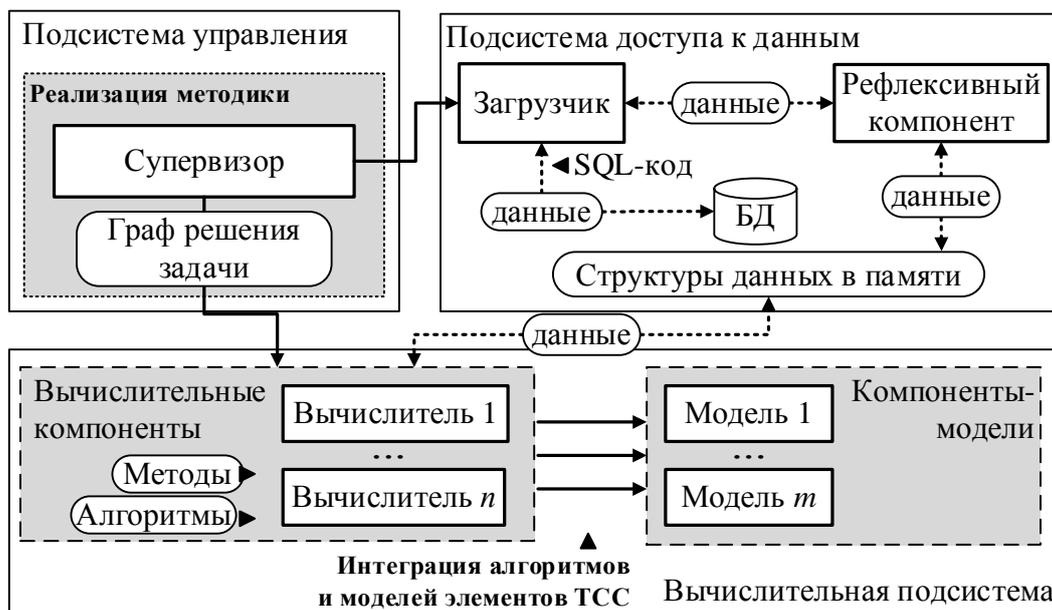


Рисунок 4 – Архитектура программной системы, получаемой в результате автоматического построения

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.

Представленная схема отражает основные положения предложенной авторами расширяемой архитектуры программного обеспечения для решения задач проектирования ТСС.

3 Инструментальная платформа

Программная система строится автоматически на основе разработанной инструментальной платформы. В качестве базового языка программирования для реализации инструментальной платформы используется Java. Его выбор обусловлен следующими преимуществами [13-15]:

- поддержка современных технологий программирования, таких как объектно-ориентированное, компонентное и функциональное программирование;
- встроенная поддержка технологий метапрограммирования (рефлексивное программирование, возможность использовать компилятор для динамического создания программных компонентов, технология Java Scripting);
- широкий набор технологий и инструментов для организации распределенных и параллельных вычислений;
- единая универсальная технология доступа к базам данных JDBC (Java Database Connectivity).

Программные компоненты, реализующие методы и алгоритмы, могут быть реализованы на любых языках программирования (С, С++, Fortran и др.). Интеграция этих компонентов осуществляется через стандартизированные интерфейсы, реализуемые за счет применения паттернов проектирования (*англ.* design patterns) [14]. Применение паттернов Command и Adapter обеспечивает универсальный интерфейс доступа к программным компонентам [12].

Структура инструментальной платформы включает следующие составные части:

- библиотеки программных компонентов для разработки программного обеспечения, предназначенного для решения задач оптимального проектирования ТСС;
- супервизор – подсистема управления построением программной системы и решением прикладной задачи;
- подсистема построения модели программной системы;
- подсистема построения программной системы на основе ее модели;
- онтологическая система как единое средство формализованного хранения знаний о теплоснабжающих системах, связанных с ними задачах и программном обеспечении, предназначенном для решения этих задач;
- программное обеспечение для работы с онтологической системой;
- библиотеки математических и графических моделей, описывающих элементы ТСС и набор стандартного оборудования.

Библиотеки программных компонентов инструментальной платформы разделены на три группы по функциональному назначению.

1. Библиотека вычислительных компонентов. Набор программных компонентов, содержащих реализации математических методов и алгоритмов для решения задач проектирования ТСС, образует библиотеку вычислительных компонентов инструментальной платформы. Компоненты этой группы разрабатываются для решения задачи многократного использования программных реализаций математических методов и алгоритмов в соответствии с следующим принципом: однажды реализованные компоненты могут многократно использоваться при создании программного обеспечения различного назначения и проведения вычислительных экспериментов при выполнении научных исследований.

2. Библиотека системных компонентов. Системные компоненты обеспечивают решение следующих задач: взаимодействие с базами данных, подготовку и экспертизу данных, формирование графического пользовательского интерфейса.

3. Библиотека управляющих компонентов. Набор программных компонентов, предназначенных для анализа состояния вычислительного процесса и принятия решения по его дальнейшему ходу. Например, эти компоненты проверяют условия сходимости итерационного процесса.

Решение прикладных задач проектирования ТСС требует организации сложного итерационного процесса, формируемого из различных подзадач, поэтому разработанная

архитектура включает специальную подсистему – *супервизор*. Эта подсистема отвечает за выполнение следующих функций:

- вызов программных компонентов, формирующих модель программной системы;
- вызов программных компонентов, отвечающих за автоматическое построение программной системы;
- управление ходом вычислительного процесса при решении прикладной задачи.

Подсистема построения модели программной системы на основе компьютерной модели ТСС, описания прикладной задачи и знаний из онтологической системы строит модель программной системы, которая представляет собой совокупность структур данных, содержащих описание создаваемого программного обеспечения.

Подсистема построения программной системы автоматически строит программную систему на основе ее модели. Эта подсистема включает программные компоненты, отвечающие за генерацию программного кода и его динамическую компиляцию, загрузку и интеграцию программных компонентов при помощи рефлексивного программирования.

В качестве единого средства для хранения формализованных знаний в рамках предлагаемого методического подхода используется *онтологическая система*. В ней хранятся знания о теплоснабжающих системах, связанных с ними задачах и программном обеспечении, предназначенном для решения этих задач. В качестве средства формализации онтологической системы в рамках предложенного методического подхода выбран язык XML (англ. eXtensible Markup Language – расширяемый язык разметки).

Модель онтологической системы S^O может быть представлена в следующем виде:

$$S^O = (O^M, O^{\Pi}, \Omega^{MB}),$$

где O^M – метаонтология; O^{Π} – множество предметных онтологий; Ω^{MB} – модель машины вывода.

Множество предметных онтологий O^{Π} описывается выражением

$$O^{\Pi} = \{O^{TCC}, O^3, O^{PO}\},$$

где O^{TCC} – онтология теплоснабжающей системы (онтология ТСС); O^3 – онтология задач; O^{PO} – онтология программного обеспечения (онтология ПО).

Метаонтология – это онтология верхнего уровня, которая специфицирует множество предметных онтологий. Она содержит базовые понятия и отношения между ними, которые используются при построении предметных онтологий.

Онтология ТСС содержит описание свойств подсистем тепловой сети, типов и параметров оборудования, иерархии структуры элементов сети, их свойств, связей между элементами, а также классификацию и описание свойств используемых математических моделей (например, замыкающие уравнения, формулы расчета сопротивлений и др.). Фрагмент онтологии ТСС представлен на рисунке 5.

Онтология задач содержит описание прикладных задач и методик их решения, описание алгоритмов верхнего уровня, перечисление параметров, являющихся исходными данными и параметров, получаемых в результате решения задачи.

Онтология ПО предназначена для хранения знаний, необходимых для автоматизации построения и использования программного обеспечения. Эта онтология содержит описание программных компонентов и их свойств, метаданные (входные и выходные параметры, описание форматов данных) и описание технологий и интерфейсов доступа к программным компонентам системы. Она содержит описание групп, на которые разделены компоненты (вычислительные, системные и управляющие). В онтологии хранятся знания о программных компонентах, входящих в каждую группу.

Программное обеспечение для работы с онтологической системой предназначено для просмотра, добавления и редактирования знаний, формализованных в онтологиях. В его состав входит набор программ для работы с онтологической системой.

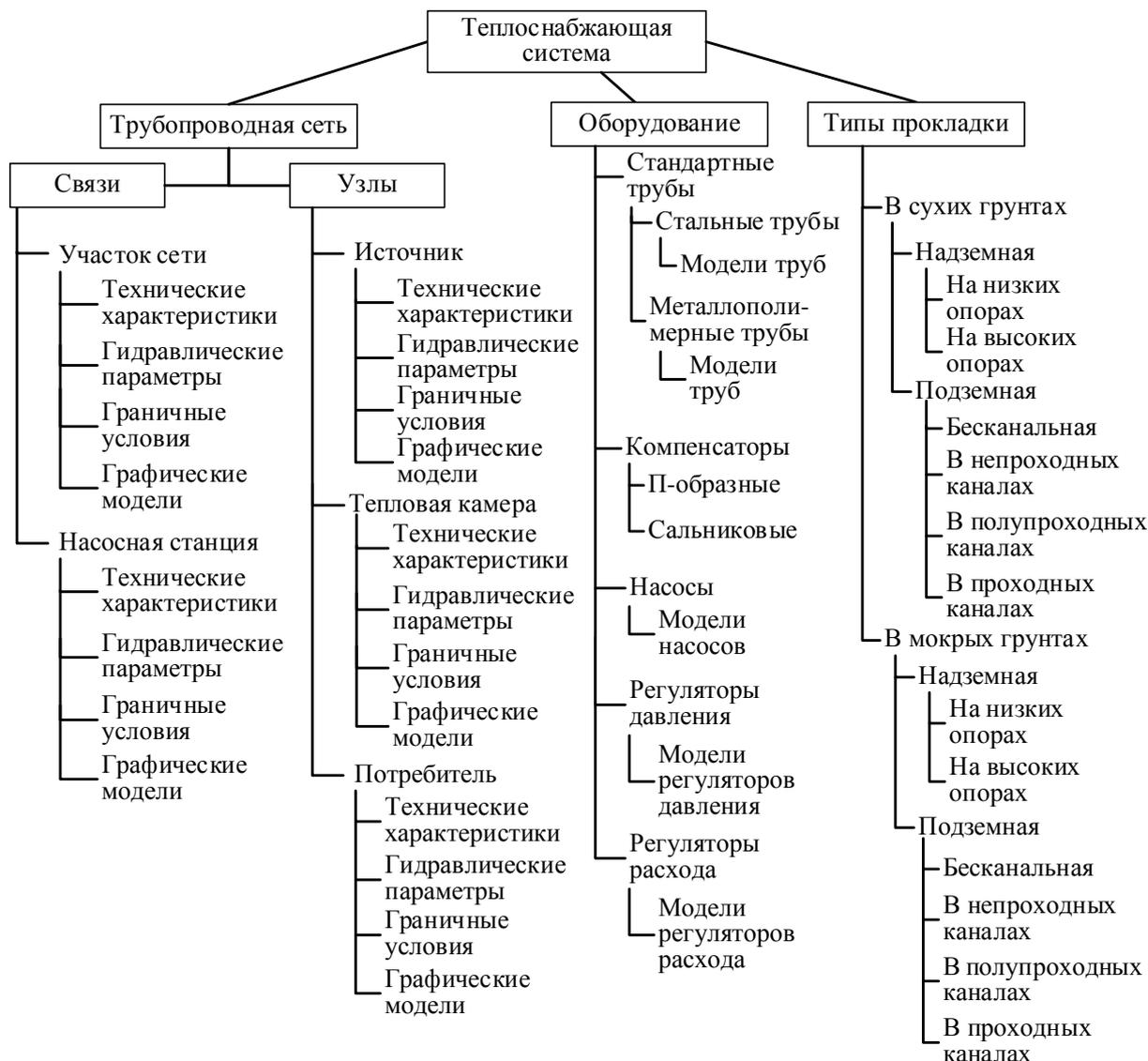


Рисунок 5 – Фрагмент онтологии теплоснабжающей системы

Библиотека математических моделей содержит математическое описание элементов ТСС, для формализованного хранения которых используется язык MathML. Для хранения библиотеки графических моделей используется векторный графический формат SVG. На основе математических и графических моделей при автоматическом построении программного обеспечения создаются программные компоненты-модели, которые используются при проведении математических расчетов и отображении компьютерной модели ТСС во время решения научных и прикладных инженерных задач.

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) [16], предназначенный для решения сложных инженерных задач оптимизации параметров тепловых сетей, многократно решаемых при проектировании ТСС [1, 2]. ПК СОСНА позволяет определять:

- «узкие места» в системе и способы их устранения;
- способы реконструкции существующих участков;
- диаметры трубопроводов новых участков сети;
- места установки и параметры насосных станций.

ПК СОСНА представляет собой сложную прикладную программную систему, которая автоматически создается в контексте решения прикладной задачи на основе компьютерной модели исследуемой ТСС, формализованного описания решаемой задачи и знаний предметной области, хранение которых организовано в виде онтологической системы. В процессе автоматического построения программной системы используется разработанная авторами инструментальная платформа, которая позволяет за счет применения современных средств метапрограммирования автоматически построить ПК и настроить на решение конкретной задачи. Широкие возможности разработанного методического подхода обеспечили осуществление декомпозиции ПК на отдельные составляющие и их увязку в виде иерархической структуры взаимодействующих друг с другом компонентов. ПК СОСНА разработан в соответствии с архитектурой, представленной на рисунке 4. Применение разработанного методического подхода при реализации ПК СОСНА обеспечило создание открытой программной системы, приспособленной для развития и сопровождения.

Применение онтологий при реализации программного обеспечения позволило получить следующие возможности:

- формализовано хранить и многократно использовать знания о подсистемах ТСС и их свойствах, задачах проектирования ТСС и применяемых при их решении программных компонентах;
- управлять процессом автоматического построения программной системы и настраивать ее на свойства конкретной ТСС и решаемую задачу;
- управлять процессом решения прикладной задачи при помощи изменения соответствующих онтологий;
- создавать модели элементов ТСС и выполнять их интеграцию в программное обеспечение при проведении научных или инженерных расчетов без изменения программного кода.

В ПК СОСНА реализованы эффективные методы оптимизации параметров ТСС, разработанные в ИСЭМ СО РАН в рамках ТГЦ. Для оптимизации разветвленных сетей реализован метод пошаговой оптимизации, разработанный на базе динамического программирования [1]. Для оптимизации кольцевых сетей реализован метод многоконтурной оптимизации (МКО), основанный на последовательном улучшении решений [1]. Важная особенность перечисленных методов состоит в том, что они позволяют полностью учесть особенности используемого дискретного набора оборудования, дискретных состояний системы и сложность математических моделей. В ПК СОСНА авторами реализована новая методика решения задач оптимизации параметров ТСС, основанная на многоуровневой декомпозиции модели тепловой сети.

С помощью ПК СОСНА проводились многовариантные расчеты достаточно сложных ТСС для получения решений по их проектированию, а также выполнялись научные

исследования по проверке работоспособности и оценке эффективности предложенных методов и алгоритмов. С помощью ПК были подготовлены и предложены технические решения по развитию ТСС Центрального и Адмиралтейского районов Санкт-Петербурга, города Братска и поселка Магистральный. На рисунке 6 приведена схема теплоснабжающей системы г. Братска и рекомендованные по результатам расчетов мероприятия по ее реконструкции.

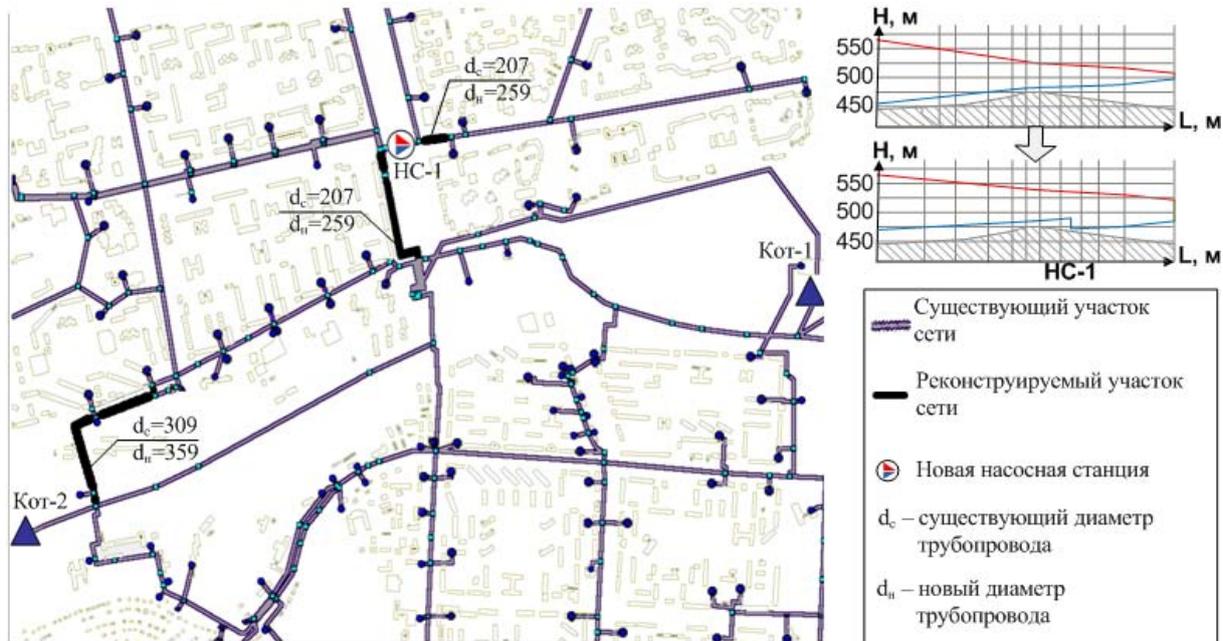


Рисунок 6 – Схема теплоснабжающей системы г. Братска с обозначенными рекомендованными мероприятиями по ее реконструкции

Результаты проведенных расчетов были использованы при подготовке технико-экономических обоснований по оптимальной реконструкции перечисленных выше и других ТСС.

Заключение

Предложенный методический подход к разработке программного обеспечения, основанный на применении концепции MDE, позволяет автоматически создавать сложные программные системы для решения научных и прикладных задач оптимального проектирования сложных технических систем. На базе этого подхода разработан ПК СОСНА для оптимизации параметров ТСС при решении задач их проектирования. ПК ориентирован на применение в научно-исследовательских, проектных и эксплуатационных организациях, занимающихся вопросами теплоснабжения. Практическое применение разработанного методического и программного обеспечения позволяет получать рекомендации по преобразованию ТСС, повышающие эффективность их работы и качество снабжения потребителей тепловой энергией.

Благодарности

Исследования выполняются при частичной финансовой поддержке Российского фонда фундаментальных исследований (грант № 13-07-00297).

СПИСОК ИСТОЧНИКОВ

- [1] **Меренков, А.П.** Теория гидравлических цепей / А.П. Меренков, В.Я. Хасилев. – М.: Наука, 1985. – 280 с.
- [2] **Стенников, В.А.** Методы комплексной оптимизации развития теплоснабжающих систем / В.А. Стенников, Е.В. Сеннова, Т.Б. Ощепкова // Энергетика. М.: Изв. РАН, 2006. №3. – С. 44–54.
- [3] **Guarino, N.** Understanding, Building, and Using Ontologies. / N. Guarino // URL: http://www.academia.edu/516503/Understanding_building_and_using_ontologies (Актуально на 12.11.2014).
- [4] **Гаврилова, Т.А.** Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский – СПб.: Питер, 2000. – 384 с.
- [5] Kogalovsky, M.R. Conceptual and ontological modeling in information systems / M.R. Kogalovsky, L.A. Kalinichenko // Programming and Computer Software. – 2009, Vol. 35, No. 5. – P. 241–256.
- [6] **Brambilla, M.** Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering #1. / M. Brambilla, J. Cabot, M. Wimmer. – Morgan & Claypool, USA, 2012. – 182 p.
- [7] **Völter, M.** Model-Driven Software Development: Technology, Engineering, Management. / M. Völter, T. Stahl, J. Bettin, A. Haase, S. Helsen. – Wiley, USA, 2006. – 446 p.
- [8] **Bartlett, J.** The art of metaprogramming, Part 1: Introduction to metaprogramming. / J. Bartlett // URL: <http://www.ibm.com/developerworks/library/l-metaprogl/> (Актуально на 22.10.2014).
- [9] **Евгеньев, Г.Б.** Интеллектуальные системы проектирования / Г.Б. Евгеньев. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. – 334 с.
- [10] **Загоруйко, Ю.А.** Онтологический подход к разработке системы поддержки принятия решений на нефтегазодобывающем предприятии / Ю.А. Загоруйко, Г.Б. Загоруйко // Вестник НГУ, серия «Информационные технологии», Том 10, Выпуск 1. Новосибирск: НГУ, 2012. – С. 121–129.
- [11] **Ворожцова, Т.Н.** Использование онтологий при моделировании программного комплекса / Т.Н. Ворожцова, С.К. Скрипкин // Вычислительные технологии, т.13, ч. I, 2008. – С. 376–381.
- [12] **Стенников, В.А.** Применение метапрограммирования в программном комплексе для решения задач схемно-параметрической оптимизации теплоснабжающих систем / В.А. Стенников, Е.А. Барактенко, Д.В. Соколов // Программная инженерия. М.: Новые технологии, 2011. №6. – С. 31–35.
- [13] **Forman, I.** Java Reflection in Action / I. Forman, N. Forman. – Greenwich: Manning Publications Co., 2005. – 273 p.
- [14] **Martin, R.C.** Agile Software Development: Principles, Patterns and Practices / R.C. Martin – Pearson Education, USA. 2002. – 529 p.
- [15] **Gosling, J.** Java Language Specification, Java SE 8 Edition / J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley. – Addison-Wesley Professional, USA, 2014. – 792 p.
- [16] **Стенников, В.А.** Применение онтологий при реализации программного комплекса для решения задач оптимального проектирования теплоснабжающих систем / В.А. Стенников, Е.А. Барактенко, Д.В. Соколов // Информационные технологии. М.: Новые технологии, 2013. №3. – С. 2–7.

USAGE OF ONTOLOGIES IN THE IMPLEMENTATION OF THE CONCEPT OF MODEL-DRIVEN ENGINEERING FOR THE DESIGN OF HEAT SUPPLY SYSTEMS

V.A. Stennikov¹, E.A. Barakhtenko², D.V. Sokolov³

Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences (ESI SB RAS), Irkutsk, Russia

¹sva@isem.sei.irk.ru, ²barakhtenko@isem.sei.irk.ru, ³sokolov_dv@isem.sei.irk.ru

Abstract

The authors suggest a methodological approach to the creation of software for heat supply system design. The approach is based on the Model-Driven Engineering (MDE) methodology and ontologies. MDE is a software development methodology, which focuses on creating and exploiting domain models. Domain models are abstract representations of knowledge and activities that govern a particular application domain. The methodological approach allow us to automatically create complex software for optimal design of heat supply systems taking into account the peculiarity of application problems and individual conditions of the development of these systems. The developed approach has been

used for the implementation of the software SOSNA intended for solving the problem of heat supply system parameter optimization.

The reported study was supported by Russian Foundation of Basic Research, research project No.13-07-00297.

Key words: *Model-Driven Engineering, metaprogramming, ontology, software, component based programming, ontological system, methodological approach, automation of programming, software architecture, software engineering, heat supply system.*

References

- [1] **Merenkov, A.P.** Teoria gidravlicheskih sistem [The Theory of Hydraulic Circuits] / A.P. Merenkov, V.Ya. Khasilev. – Moscow: Nauka, 1985. – 280 p. (In Russian).
- [2] **Stennikov, V.A.** Metody kompleksnoj optimizacii razvitiya teplosnabzhajushhih sistem [Methods of complex optimization for heat supply system development] / V.A. Stennikov, E.V. Sennova, T.B. Oshchepkova // Jenergetika. Moscow: Izvestija RAN, 2006, No.3. – P.44-54 (In Russian).
- [3] **Guarino, N.** Understanding, Building, and Using Ontologies / N. Guarino // URL: http://www.academia.edu/516503/Understanding_building_and_using_ontologies (Valid on 12.11.2014).
- [4] **Gavrilova, T.A.** Bazy znaniy intellektual'nyh sistem [Knowledge base of intelligent systems] / T.A. Gavrilova, V.F. Horoshevskij. – Saint Petersburg: Piter, 2000. – 384 p. (In Russian).
- [5] **Kogalovsky, M.R.** Conceptual and ontological modeling in information systems / M.R. Kogalovsky, L.A. Kalinichenko // Programming and Computer Software. – 2009, Vol. 35, No. 5. – P. 241–256.
- [6] **Brambilla, M.** Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering #1. / M. Brambilla, J. Cabot, M. Wimmer. – Morgan & Claypool, USA, 2012. – 182 p.
- [7] **Völter, M.** Model-Driven Software Development: Technology, Engineering, Management. / M. Völter, T. Stahl, J. Bettin, A. Haase, S. Helsen. – Wiley, USA, 2006. – 446 p.
- [8] **Bartlett, J.** The art of metaprogramming, Part 1: Introduction to metaprogramming. / J. Bartlett // URL: <http://www.ibm.com/developerworks/library/l-metaprog1/> (Valid on 22.10.2014).
- [9] **Evgenev, G.B.** Intellektual'nye sistemy proektirovaniya [Intelligent systems design] / G.B. Evgenev. – Moscow: Izd. MGTU im. N.Je. Baumana, 2009. – 334 p. (In Russian).
- [10] **Zagorulko, Y.A.** Ontologicheskij podhod k razrabotke sistemy podderzhki prinjatija reshenij na neftegazodobyvajushhem predpriyatii [An ontological approach for the development of a decision support system for an oil production enterprise] / Y.A. Zagorulko, G.B. Zagorulko // Vestnik NGU, series «Informacionnye tehnologii», Volume 10, Issue 1. Novosibirsk: NSU, 2012. – P. 121-129. (In Russian).
- [11] **Vorozhcova, T.N.** Ispol'zovanie ontologij pri modelirovanii programmogo kompleksa [The use of ontologies for modeling software system] / T.N. Vorozhcova, S.K. Skripkin // Vychislitelnye tehnologii, Vol.13, part I, 2008. – P. 376-381. (In Russian).
- [12] **Stennikov, V.A.** Primenenie metaprogrammirovaniya v programmnom komplekse dlja reshenija zadach shemno-parametricheskoj optimizacii teplosnabzhajushhih sistem [Metaprogramming in the software for solving the problems of heat supply system schematic and parametric optimization] / V.A. Stennikov, E.A. Barakhtenko, D.V. Sokolov // Programmaya ingeneriya. Moscow: Novye tekhnologii, 2011. No.6. – P.31-35. (In Russian).
- [13] **Forman, I.** Java Reflection in Action / I. Forman, N. Forman. – Greenwich: Manning Publications Co., 2005. – 273 p.
- [14] **Martin, R.C.** Agile Software Development: Principles, Patterns and Practices / R.C. Martin – Pearson Education, USA, 2002. – 529 p.
- [15] **Gosling, J.** Java Language Specification, Java SE 8 Edition / J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley. – Addison-Wesley Professional, USA, 2014. – 792 p.
- [16] **Stennikov, V.A.** Primenenie ontologij pri realizacii programmogo kompleksa dlja reshenija zadach optimal'nogo proek-tirovaniya teplosnabzhajushhih sistem [Ontologies application for implementation of the software for solving optimal design problems of heat supply systems] / V.A. Stennikov, E.A. Barakhtenko, D.V. Sokolov // Informacionnye tehnologii. Moscow: Novye tekhnologii, 2013. No.3. P.2-7. (In Russian).

Сведения об авторах



Стенников Валерий Алексеевич, 1954 г. рождения. Доктор технических наук, профессор, заслуженный деятель науки РФ. Заместитель директора по науке ИСЭМ СО РАН, Иркутск. Получил степень кандидата технических наук (1985 г.) и доктора технических наук (2002 г.) в ИСЭМ СО РАН. Область научных интересов: методология, математические модели и методы для обоснования развития систем теплоснабжения с учетом надежности и управляемости; основополагающие тенденции и закономерности развития систем теплоэнергетики и теплоснабжения, энергоэффективных технологий и оборудования, развиваемых в России и за рубежом; методы и алгоритмы расчета экономически обоснованных дифференцированных тарифов на тепловую энергию. Стенников В.А. является автором и соавтором более чем 270 научных работ.

Stennikov Valery Alekseevich was born in 1954 in Russia, PhD, Professor, Doctor of Science. He is the Deputy director of Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia. Valery Stennikov received his degree of Candidate of Technical Sciences at Energy Systems Institute (Irkutsk) in 1985 and Doctor of Technical Sciences at Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences (Irkutsk) in 2002. His research interests include: The methodology, mathematical models and methods for a development substantiation heat supply systems taking into account reliability and controllability requirements; basic tendencies and laws of development of heat supply system, power effective technologies and the equipment; energy saving; methods and algorithms of calculation of economically well-founded differentiated tariffs for thermal energy. Professor Stennikov is the author and co-author of more than 270 scientific papers.



Барахтенко Евгений Алексеевич, 1982 г. рождения. Окончил Иркутский государственный технический университет в 2005 г. Получил степень кандидата технических наук (2011 г.) в Иркутском государственном техническом университете. Старший научный сотрудник ИСЭМ СО РАН. Область научных интересов: методическое и программное обеспечение для решения проблем оптимального развития и реконструкции трубопроводных систем. Барахтенко Е.А. является автором и соавтором более 30 научных работ.

Barakhtenko Evgeny Alekseevich was born in 1982 in Russia. He graduated from Irkutsk State Technical University in 2005. Evgenie Barakhtenko received his degree of Candidate of Technical Sciences at Irkutsk State Technical University, Irkutsk, Russia in 2011. He is Senior Researcher at Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia. His research interests include: methodical bases and software for

solving the problems of optimal development and reconstruction of pipeline systems. Evgenie Barakhtenko is the author and the coauthor of more than 30 scientific papers.



Соколов Дмитрий Витальевич, 1984 г. рождения. Получил степень кандидата технических наук (2013 г.) в ИСЭМ СО РАН. Научный сотрудник ИСЭМ СО РАН. Область научных интересов: методическое и программное обеспечение для решения проблем реконструкции теплоснабжения систем, методы оптимизации, разработка быстродействующих алгоритмов и программного обеспечения. Дмитрий Соколов является автором и соавтором более 20 научных работ.

Sokolov Dmitriy Vitalyevich was born in 1984 in Irkutsk. He is Researcher at Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia. Dmitriy Sokolov received his degree of Candidate of Technical Sciences at Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences in 2013. Subject of his research is the methodical support and software for solving of problems of heat supply systems reconstruction, optimization methods, development of high-speed algorithms and software. Dmitriy Sokolov is the

co-author of more than 20 scientific papers.